1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 14.Aug.02	3. REPORT TYPE AND DATES COVERED THESIS

4. TITLE AND SUBTITLE
MODEL PREDICTIVE CONTROL FOR ASCENT LOAD MANAGEMENT OF A REUSABLE LAUNCH VEHICLE

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT MARTIN ANDREW A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER

CI02-124

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

20020829 038

14. SUBJECT TERMS

	15. NUMBER OF PAGES 190
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Model Predictive Control for Ascent Load Management of a Reusable Launch Vehicle

by

Andrew Allen Martin

B.S. Astronautical Engineering
United States Air Force Academy, 2000

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

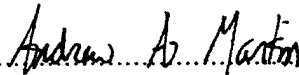Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

© 2002 Andrew Allen Martin. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly
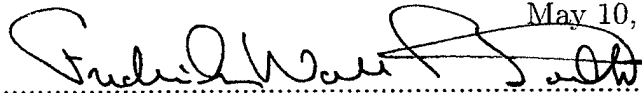paper and electronic copies of this thesis document in whole or in part.

Signature of Author ....
Department of Aeronautics and Astronautics
May 10, 2002
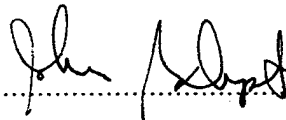
Certified by ....
Frederick W. Boelitz
The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor
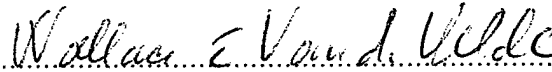
Certified by ....
John J. Deyst
Professor of Aeronautics and Astronautics, MIT
Thesis Advisor

Accepted by ....
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics, MIT
Chairman, Committee on Graduate Students

Model Predictive Control for Ascent Load Management of a Reusable Launch Vehicle

by

Andrew Allen Martin

Abstract

During the boost phase of ascent, winds have a significant impact on a launch vehicle's angle of attack, and can induce large structural loads on the vehicle. Traditional methods for mitigating these loads involve measuring the winds prior to launch and designing trajectories to minimize the vehicle angle of attack (α). The current balloon-based method of collecting wind field information produces wind profiles with significant uncertainty due to the inherent time delays associated with balloon measurement procedures. Managing the mission risk caused by these uncertain wind measurements has always been important to control system designers. This thesis will describe a novel approach to managing structural loads through the combination of a Light Detection and Ranging (LIDAR) wind sensor, and Model Predictive Control (MPC). LIDAR wind sensors can provide near real-time wind measurements, significantly reducing wind uncertainty at launch. MPC takes full advantage of this current wind information through a unique combination of proactive control, constraint integration and tuning flexibility. This thesis describes the development of two types of MPC controllers, as well as a baseline controller representative of current control methods used by industry. A complete description of Model Predictive Control theory and derivation of the necessary control matrices is included. The performance of each MPC controller is compared to that of the baseline controller for a wide range of wind profiles from both the Eastern and Western U.S. Test Ranges. Both MPC controllers are shown to provide reductions of greater than 50% in α, $Q\alpha$ and structural bending moments. In addition, the effects of wind measurement delays and uncertainty on the performance of each controller are investigated.

Thesis Supervisor: Frederick W. Boelitz
Title: Staff, The Charles Stark Draper Laboratory, Inc.

Thesis Advisor: John J. Deyst
Title: Professor of Aeronautics and Astronautics, MIT

Acknowledgments

I would like to express my sincere appreciation to everyone who has played a part in the completion of this thesis during the past two years of my life here in Boston. I have been truly blessed with a wonderful circle of friends, family, coworkers and supervisors during my time here at Draper Labs and MIT, and I can't thank everyone enough.

I would like to say thanks to my supervisors. First to Fred Boelitz, who has been an outstanding boss during my time at Draper. Not only has he served as a constant voice of support, experience and guidance during the entire thesis process, but he has also constantly pushed me to excel in my research. Secondly, I would like to thank Dr. John Deyst, my MIT thesis advisor, for his instrumental help in completing my thesis. I would also like to say thanks to Chris Gibson and Piero Miotto, for always being willing to take time out of their busy schedules to give me advice or help.

Thanks also to my Air Force Academy advisor, Dr. Paul Vergez, and Mr. Chuck Moon, for their input on my thesis. The outside viewpoints were invaluable. I would also like to thank my roommates, Rob LePome, Josh McConnell, Laura Durham and Jason Sickler for putting up with me for the last year. You guys made my stay out in Somerville quite enjoyable. Thanks to my Draper buddies also, Josh and Rob again, as well as Ted Dyckman, Kathy Benzin and Trent Yang.

I would like to say thanks to my wonderful parents and family, both nuclear and extended, who have offered so much support and prayer during the past years, through all my academic pursuits. Thanks to all the MIT and PSC friends that I have made in the last two years. Thanks especially to my small group for being so supportive during this process. My life has definitely been brighter because of you.

Thanks also to all my instructors from the U.S. Air Force Academy's Department of Astronautics. I could not have asked for a better engineering education than the one that I received from you, and for that I am extremely grateful.
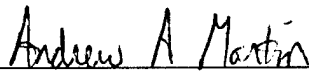
Finally and most importantly, I would like to praise Jesus Christ, my Lord and Savior, for his constant presence and support in my life during all these years. His blessings have truly been without bound.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Independent Research and Development #18558, Next Generation Guidance and Control.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Permission is hereby granted by the author to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

_____

Andrew A. Martin, 2Lt., USAF   May 10, 2002

Contents

[This page intentionally left blank.]

List of Figures

12

14

[This page intentionally left blank.]

List of Tables

[This page intentionally left blank.]

Chapter 1

Introduction

During the boost phase of ascent, winds have a significant impact on a launch vehicle's angle of attack, and can induce large structural loads on the vehicle. These structural loads have a direct impact on launch safety, and are monitored closely during preflight launch preparations. Over the past 40 years, launch procedures have evolved which attempt to quantify and minimize the risks that winds induce upon launch attempts. These procedures usually involve several pre-launch balloon releases. The balloons are monitored by radar, which record their rise through the atmosphere and the associated winds that are present at various altitude levels. Because of the relative expense of these balloons, as well as the personnel required to launch and track them, few are used for each launch, resulting in dated wind field information which contains significant uncertainty. The time between wind state observation and vehicle launch is known as the lapse time.

To deal with this uncertainty, two possible courses of action have been taken in the past. The first is to directly address the issue of wind knowledge certainty. This is done by enforcing conservative wind requirements on the launch decision. This approach seeks to limit mission risk by ensuring that the possibility of an unsafe wind situation developing within the lapse time is below a certain threshold. Thus, extremely limiting wind requirements are implemented, which often cause launch delays or cancellations due to marginal or quickly changing wind conditions.

The second possible course of action is to implement load management procedures

or systems onboard the vehicle, so that it can manage wind changes during flight in real-time. Several advances have been made in this area throughout the past decade. However, these changes have only partially addressed the issue of wind uncertainty.

This thesis will address both of these methods of dealing with uncertainty by proposing and evaluating a load relief system which consists of a predictive controller using Model Predictive Control (MPC), working in conjunction with a real time wind sensor, known as a Doppler Light Detection and Ranging (LIDAR) sensor, which can provide accurate and timely wind knowledge to the controller. This sensor, which uses reflected laser pulses to measure air movement in the atmosphere could be located on the vehicle itself, on the ground near the launch site, or on an aircraft flying over the launch site. Although the technical specifics of this type of device are beyond the scope of this thesis, wind sensors with these capabilities are currently under development in the private sector, and will be available in a matter of years.

Using real time wind knowledge supplied by the LIDAR wind sensor, the MPC controller is able to anticipate wind disturbance changes, and position the vehicle attitude in flight so as to minimize the structural impact of wind changes on the vehicle. This capability significantly expands the acceptable launch envelope of the vehicle, resulting in fewer launch delays and scrubs. In addition, this technology could be incorporated into the design of future launch vehicles, possibly allowing structural mass savings and equivalent increases in payload capacity in response to relaxed load bearing requirements.

1.1   Problem Motivation

As stated previously, wind knowledge uncertainty causes excessive conservatism in the launch planning process. This conservatism is responsible for many delays and cancellations during questionable weather conditions. Data covering launch operations from both the Eastern Test Range (ETR) at Kennedy Space Center, FL, and the Western Test Range (WTR) at Vandenburg AFB, CA, during the last decade shows that nearly half of all launch cancellations were due to high altitude winds. Previous

20

industry research efforts have shown that launch delays can cost the customer an average of $500,000 per day, simply for pad access costs. This does not take into account the costs associated with the delays that other payloads might experience due to prior launch delays.

Although the problem of wind uncertainty has always existed, the technology to implement the proposed load relief solution has only recently become available. LIDAR wind sensors with enough power to monitor winds up to altitudes of 30 km are currently being developed. In addition, the computing power required to enable real-time, high-bandwidth applications of the computationally intensive MPC approach have only become available within the past few years. Prior to that, MPC was widely used by the process control industry, where requirements could be satisfied with smaller bandwidths and slower cycle times. Continuing growth in processor speeds should allow MPC to be applied to an increasingly large segment of flight control applications in the future. Because of MPC's predictive nature, it is able to fully utilize current wind profile information, which can only be provided by a real-time wind sensor, such as a Doppler LIDAR. Thus, these two technologies naturally compliment each other as an integrated solution to the launch uncertainty problem.

1.2 Overview

The overall goal of this research is to showcase the advantages and disadvantages of the proposed MPC/LIDAR load-relief system, as compared to a traditional load-relief system using traditional wind measurement techniques. The included results will demonstrate some of the significant potential that MPC and Doppler LIDAR may have for contributing to increased launch probability. In addition, the MPC/LIDAR load-relief system will be compared with a traditional load-relief system coupled with a Doppler LIDAR wind sensor. This comparison will highlight the benefits that MPC alone can offer towards increasing launch capability.

The models used in these comparisons are identical, except for the individual controller integrated into each model. Each vehicle model is a highly accurate, 6

Degree-of-Freedom (6DOF) non-linear representation of the Kistler Aerospace Corporation (KAC) K-1 reusable launch vehicle. Each controller guides a vehicle model along trajectories generated with a preflight trajectory generator. Each simulation is subject to a set of actual wind profiles which were recorded at either the ETR or the WTR. Over 3000 wind profiles were processed for this research.

1.3 Results Portability

While the model employed for this research represented the K-1 reusable launch vehicle, the results of this thesis are certainly applicable to a wide range of launch vehicles, both reusable and expendable. The dynamics of most launch vehicles depend on a similar set of vehicle parameters, which were included in the K-1 vehicle model. Although specifics such as the location of the LIDAR wind sensor could vary with vehicle, the general result trends of this research should be applicable to most current launch vehicle platforms.

1.4 Thesis Preview

Following the introductory chapter, this thesis is organized as follows:

Chapter 2 presents the K-1 vehicle model framework, including the assumptions and simplifications which were made for this research. Other model details, such as coordinate frames, mass properties and plant identification are discussed.

Chapter 3 presents the mathematical background of MPC controller theory, as well as a verification of the MPC controller used for this research. Additional features of MPC are also discussed, such as constraints and slack variables.

Chapter 4 presents the architecture development process which was used to choose the two MPC control approaches used in this research from an original set of five MPC control architectures.

Chapter 5 presents the wind models used in this research, as well as the preprocessing which was required before the wind data could be used by the simulations.

In addition, time lapse wind processing and the Trajectory Generator are discussed.

Chapter 6 presents the common PID (Proportional-plus-Integrator-plus-Derivative) controller layout, which was used in all three simulations in some form. The baseline controller design process is discussed in detail.

Chapter 7 presents the MPC controller parameter development process. This includes parameter selection methods for input and output weighting matrix values, as well as projection and control horizons, prediction step sizes, controller rates and J_M matrices.

Chapter 8 presents the results from the comparison of the MPC controllers and the baseline controller for a range of wind profiles and time lapse scenarios.

Chapter 9 summarizes the findings of this research and concludes with lessons learned and recommendations for future research.

A comprehensive list of symbols, acronyms and conventions is included in Appendix A.

[This page intentionally left blank.]

Chapter 2

Vehicle Model

2.1 Model Framework

Traditionally, vehicle simulations are constructed in either the FORTRAN, C, or C++ software languages as collections of stand-alone executable files. Although this method produces simulations with fast execution times, quickly modifying or adjusting the simulations becomes increasingly difficult with increased size. The cost required to develop the input/output infrastructure needed to support rapid engineering analysis and design is substantial. One advantage of stand-alone simulations, however, is the supply of reliable functions that have been previously coded and tested in other programs, such as standard atmosphere models or gravity models.

The standard modelling and simulation tool employed today is SIMULINK, by MathWorks, Inc. In addition to providing a graphically driven user interface with strong input/output support, the software package can also work in unison with MathWorks' other main software product, MATLAB. MATLAB provides basic linear algebra functionality along with other tools for designing control systems and displaying signals. The tools are provided in the form of software toolboxes that can be purchased separately from MathWorks.

The ideal simulation environment would couple the rapid simulation development capabilities supplied by Simulink, with the speed supplied by stand-alone C based functions. Fortunately, Simulink and MATLAB provide an open software interface

25

that allows custom designed code to be executed from within its environment.

A Simulink S-Function (system-function) is a computer language description of a Simulink function block. S-Functions can be written in MATLAB, C, C++, Ada, or FORTRAN. S-Functions use a special calling syntax that enables custom designed functions to interact with Simulink's equation solvers. Typically, a programmer creates a *wrapper* function that calls a separate model function. The wrapper function contains all of the required application programming interface functions required by the Simulink environment, while the model function contains the actual executable code. In this way, the model function (e.g. 1976 US Standard Atmosphere) is neatly isolated from Simulink, thereby making it easily portable to other simulation environments.

2.2 Non-Linear Vehicle Model

For this thesis, all of the dynamic models that collectively describe the motion of the Kistler K-1 are coded as separate C S-Functions. These functions are integrated into Simulink by calling them from their corresponding wrapper functions, identified by the naming syntax $filename\_SMLK.c$. Hereafter, all S-Functions will be referred to simply by their filenames. All of the S-Functions are compiled into dynamic link libraries ($.dll$) on a PC using Microsoft's Visual C++ Studio version 6.0. The vector and transformation matrix naming conventions used in this thesis are detailed in Appendix A. A more complete list of the equations contained in each S-Function can be found in Appendix B. The critical dynamic models used to describe the motion of any rigid body vehicle travelling about a spherical earth include:

- **accel** : Computes the $\dot{\vec{v}}_b^i\big|_I$ as well as the $\dot{\vec{\omega}}_b^i\big|_B$. The primary inputs to the model are the vehicles mass properties and the forces and torques acting on the body.

- **eulerRates** : Computes the time rate of change of the yaw (ψ), pitch (θ), and roll (ϕ) euler angle sequence. The input is the $\dot{\vec{\omega}}_b^i\big|_B$ computed by the **accel**

26

function and the current ψ, θ, ϕ, euler angle sequence.

- **envGMST** : Computes transformation matrices T_i^e and T_e^i. The inputs to this model are time and the rotation rate of the earth.

- **xformEuler** : Computes transformation matrices T_b^i, T_i^b, T_b^e, T_e^b, T_e^{lg}, T_{lg}^e, T_b^{lg}, and T_{lg}^b. These transformations are computed using the current vehicle latitude, longitude, attitude and T_e^i.

- **velocityOmega** : Computes various velocities and angular rates of the body with respect to a variety of frames. These include $\vec{\omega}_e^i\big|_B$, $\vec{\omega}_b^i\big|_I$, $\vec{\omega}_b^{lg}\big|_B$, $\vec{\omega}_b^e\big|_B$, $\vec{v}_b^i\big|_B$, $\vec{v}_b^e\big|_B$, $\vec{v}_b^e\big|_E$, $\vec{v}_b^e\big|_{LG}$, $\vec{v}_a^e\big|_B$, $\vec{v}_b^a\big|_B$. The inputs to the model include the vehicle longitude, $\vec{\omega}_b^i\big|_B$, T_i^b, T_b^e, T_b^{lg}, $\vec{v}_b^i\big|_I$, $\vec{p}_b^i\big|_I$, $\vec{v}_a^e\big|_{LG}$ and the rotation rate of the earth.

- **gravityShepperd** : Computes the force vector on the body due to gravity, expressed in the body frame, $\vec{F}_g\big|_B$, the gravity vector in both the earth and body frames, $\vec{v}_g\big|_E$ and $\vec{v}_g\big|_B$ respectively, and the weight of the vehicle. The inputs to the model are the vehicle mass, the T_e^b and the $\vec{p}_b^e\big|_E$

- **aeroParameters** : Computes the angle of attack (α), sideslip angle (β), total angle of attack (α^T), aerodynamic roll angle (ϕ^*), airspeed, dynamic pressure (Q), and mach number based on the $\vec{v}_b^a\big|_B$, local air density (ρ), and speed of sound (SoS).

- **atmosphereUS76** : Computes the local pressure, temperature, density, and speed of sound as a function of the local radius of the Earth and the altitude of the vehicle above mean sea level.

- **xyz2lat_lon_alt** : Computes the latitude, longitude, altitude and the local radius of the earth, re, as a function of the $\vec{p}_b^i\big|_E$. Parameters include the equatorial earth radius, $re0$, and the earth flattening constant, k.

In order to uniquely define the K-1, the following additional models are required:

- **engine** : Computes the force and torque vectors produced by the rocket engines expressed in the body frame. The inputs to the model are the location of the engine hinge point in the body frame, the local air pressure, the current throttle setting [0-1], the vehicle CG location, and the pitch and yaw gimbal angles with respect to the body frame, δ_p and δ_y.

- **aeroK1Boost** : Computes the aerodynamic force and torque vectors based upon the current Q, α^T, ϕ^*, CG location, and mach number.

The full non-linear K-1 simulation is comprised of approximately fifteen S-Functions linked together to form a comprehensive 6 Degree-of-Freedom rigid body simulation. A collection of pictures and descriptions of all 6DOF models used in this research is included in Appendix E. Due to the flexibility of the Simulink environment, any number of continuous integrators can be chosen. ($rk4$ was typically used with a sample time of $\triangle t = 0.02\,sec$) depending upon the speed and fidelity required. All work described in this thesis was completed using Simulink version 4 and MATLAB version 6.1.0.450 release 12.1.

2.2.1 Standard Coordinate Frame Descriptions

The full simulation uses the four standard frames found in space system simulations. All frames are orthonormal and right-handed. Figures 2-1 and 2-2 illustrate these coordinate frames.

- The earth-centered inertial frame (ECI), commonly called the Inertial Frame. This frame, with axes I, J and K, is fixed at the earth's center, with the I axis pointing towards the vernal equinox, the J axis perpendicular to the I axis within the equatorial plane, and the K axis completing the right-hand system, pointing out the North Pole, along the earth's axis of rotation.

- The earth-centered rotating frame (ECR), commonly called the Earth Frame. This frame is similar to the ECI frame, except that it rotates with the earth,

with the primary axis aligned with a fixed meridian, normally the Greenwich Meridian.

- The local geographic frame, commonly called the *NED* or *LG* Frame. This frame is fixed at some point on the earth's surface, with the primary axis pointing North, the second pointing East and the third axis, completing the right-handed system, pointing Down.

- The body-fixed frame, commonly called the Body Frame. This frame is fixed at some point on the body, usually the CG. The primary axis (X) points out the nose of the vehicle, with the second axis (Y) perpendicular to the primary axis, usually out the right side of the vehicle. The third axis (Z) completes the right-handed system, normally pointing out the bottom of the vehicle.
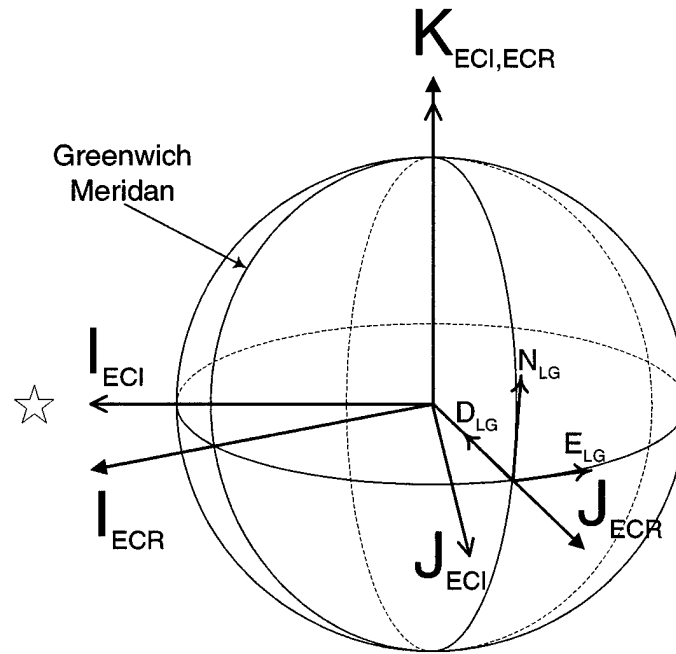
Figure 2-1: ECI, ECR and NED Coordinate Frames
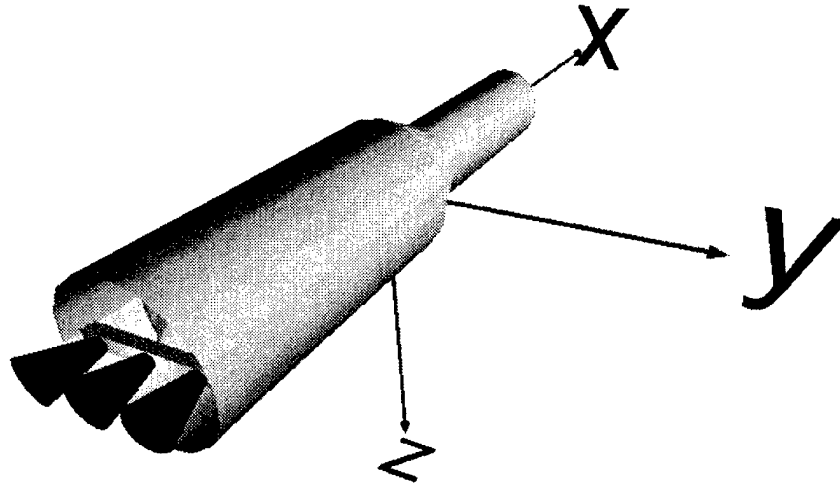
Figure 2-2: Body Coordinate Frame

2.2.2 Assumptions and Simplifications

Several assumptions and simplifications were made in constructing the non-linear vehicle model. These are listed below and on the following pages, along with their justifications:

- The Earth is modelled as a flat surface with an X, Y, Z inertial reference frame. Only this frame and the body frame are required. Because this research is constrained to the relatively short (134 sec) boost phase of ascent, the vehicle travels over a very small percentage of the earth's surface. Therefore, any errors resulting from differences in navigating through a spherical Earth environment as opposed to a flat Earth environment can be shown to be negligible.

- The vehicle is assumed to be a rigid body. Effects due to liquid fuel slosh and structural bending are assumed to be higher order and will not impact the performance of the proposed guidance system. This assumption is based on previous analysis of the K-1 which shows that the first structural bending mode occurs at approximately 18.5 rad/sec, and the first slosh mode occurs

at approximately 2.3 rad/sec [3]. To capture the real world effects that these properties would have on system bandwidth, the control system design for this research was limited to between 1 and 2 rad/sec, a typical range for a launch vehicle.

- Engine gimbal dynamics are assumed to be linear and second order in nature. Frequency domain design specifications for the K-1 rocket gimbal system show that this assumption is sufficient to model the system's expected response.

- Tail-Wags-Dog (TWD) effects are excluded from the simulation. TWD refers to the dynamic effects due to the movement of the main engine nozzle which is attached to the aft end of the vehicle body. Previous analysis of the K-1 has shown that the frequency of the TWD zero is much higher than the bandwidth of the proposed control system, at approximately 30 rad/sec [3]. As with body bending, this is assumed to be a higher order effect that will not influence the performance of the proposed load relief system.

- Any transportation lags associated with on-board processing are assumed to be negligible. For the K-1, this is not necessarily true. However, the effect that computation lags may have on system performance is acknowledged by limiting the bandwidth of the proposed control system as previously stated.

- The K-1 has three Russian NK-33 booster engines, each delivering approximately 395,000 *lbf* of thrust, when measured in the vacuum of space. Pitch and yaw control is achieved by simultaneously gimballing all three engines in the same direction. Roll control is achieved by differentially gimballing the outboard engines. For this research, all three engines are represented as a single, central engine of equivalent thrust. Pitch and yaw control methods remain unchanged. However, because the simulation vehicle has only one engine, roll control is not possible. In all other respects, a single engine can effectively simulate the same dynamic effect as three separate engines.

- Although the real K-1 has aerodynamic damping forces, these are not modelled

31

in the simulation. For the K-1, these forces are small and their tendency is to add robustness to the control design. Eliminating this effect adds a small measure of conservatism to the analysis.

- The actual K-1 cross products of inertia, while not exactly zero, are so small that they can safely be assumed to be equal to zero without impacting the validity of the vehicle simulation. The average I_{xy} inertia, for example, is only 2.7% of the average roll inertia, I_{xx} and 0.03% of the average pitch inertia, I_{yy}.

- The center of gravity of the body is assumed to move as a function of the vehicle mass, and its motion is constrained to the body x-axis. This assumption is valid because the actual off-axis component of the CG location is extremely small, on the order of 1 inch for a vehicle which is approximately 115 feet in length and 22 feet in diameter.

- Feedback of angular rate, attitude, and position into the control system is assumed to be perfect. Although all measurement devices produce biased and noisy signals, the issues caused by these inaccuracies are beyond the scope of this thesis. In addition, because all systems will be affected in similar ways by signal noise, the comparison framework of this thesis will act to minimize the impact of this noise on the relative performance results of the baseline and proposed load relief guidance systems.

- All trajectories used in this research are solely pitching trajectories, and include no non-zero yaw or roll trajectory components. This allows traditional control methods to be used for yaw plane control, and negates the need for roll control.

- All winds are along the inertial frame Z axis, as shown in Figure 2-3. There are no lateral or vertical components to wind disturbances.

- Because wind disturbances only occur in the pitch plane, all research and analysis was conducted solely in this plane.

- Winds are assumed to be constant at any position at a given altitude.

32

- All simulation wind profiles are assumed to accurately represent actual wind profiles, although they were gathered from balloon measurements, which have been shown to have a slight dampening effect on wind measurements.

2.3 Simplified Flat-Earth Non-Linear Vehicle Model

As previously stated, to reduce complexity, increase simulation speed and provide for an easier to understand and operate framework, the simulation environment was converted from a Round-Earth to a Flat-Earth environment. Only two coordinate frames were retained, as shown in Figure 2-3. The S-Functions *envGMST* and *xyz2lat_lon_alt* were removed. In addition, the S-Functions *xformEuler* and *velocityOmega* were rewritten to eliminate all unnecessary calculations resulting from the simulation environment transition.

2.3.1 Coordinate Frame Descriptions

The simplified vehicle simulation contains two frames, the inertial frame and the body frame. The inertial frame is fixed at the launch site and follows standard right-hand, orthonormal conventions. The X axis points up, and is perpendicular to the Y and Z axes, which lie in the plane of the Earth's surface, following the right hand rule. The second frame is the body frame, which is fixed to the body with its origin at the vehicle's CG, in the traditional *nose, right wing, down* configuration as described previously. Thus, the two frames can begin the flight co-located and aligned. A picture of the inertial and body frames is included in Figure 2-3. Figure 2-4 describes the variables, angle of attack (α), pitch (θ) and flight path angle (γ). $V_b^a\big|_B$ represents the velocity vector of the body, with respect to the air. This notation convention is explained in more detail in Appendix A.
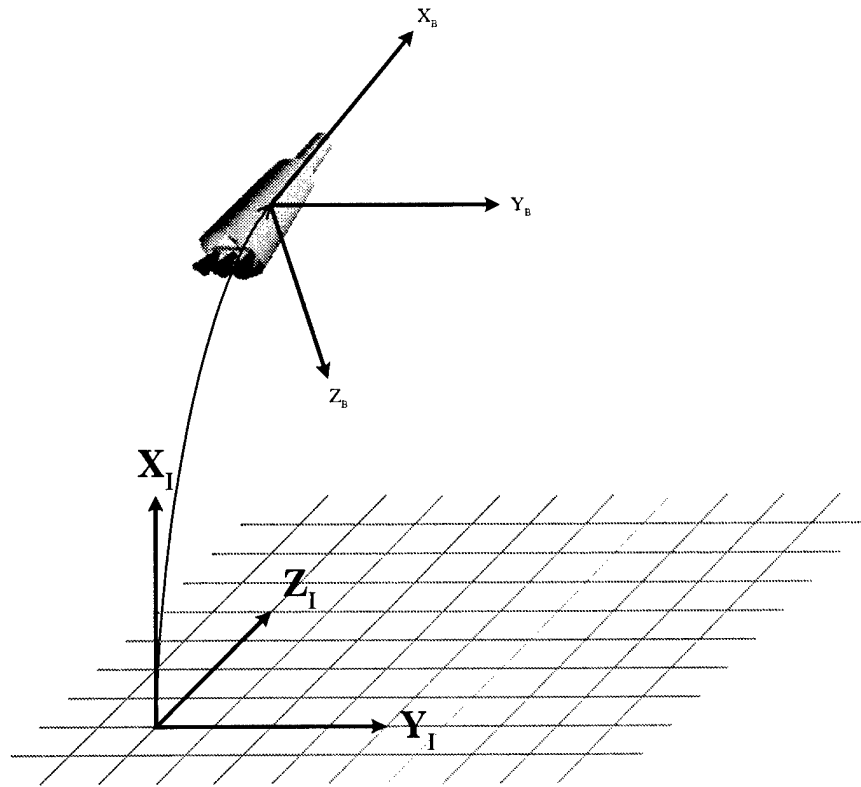
Figure 2-3: Simulation Coordinate Frames
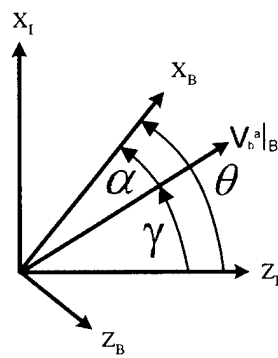
Figure 2-4: Description of Aerodynamic Variables

2.3.2   Vehicle Mass Properties

The vehicle inertia tensor and CG location information used in this simulation are representative of a Kistler K-1. Table 2.1 contains the actual values used in the vehicle simulation. Although not shown, the magnitudes of the off-axis inertia tensor components and the Y and Z CG positions are quite small, as stated in the Assumptions and Simplifications section. In addition, because all analysis is done in the pitch plane, and the vehicle has virtually no roll or yaw inputs during flight, off-axis inertia components would not significantly affect performance, regardless of their size. The CG position is measured from a point fixed 1000 inches behind the vehicle separation plane, where the two stages of the K-1 connect.

	Mass (slugs)	CGx (ft)	Ixx (slug-ft^2)	Iyy	Izz
Launch	26243	73.53	212,407	19,183,122	19,192,511
90% Fueled	24825	74.17	208,756	18,939,892	18,949,259
80% Fueled	23407	74.98	205,103	18,602,901	18,612,290
70% Fueled	21989	75.98	201,450	18,171,523	18,180,934
60% Fueled	20571	77.19	197,796	17,631,988	17,641,377
50% Fueled	19153	78.65	194,140	16,960,056	16,969,467
40% Fueled	17735	80.44	190,482	16,123,094	16,132,526
30% Fueled	16317	82.63	186,821	15,074,392	15,083,824
20% Fueled	14899	85.33	183,158	13,746,394	13,755,869
10% Fueled	13481	88.74	179,490	12,034,330	12,043,805

Table 2.1: Vehicle Simulation Mass Properties

2.3.3   Plant Creation

A separate plant ID program was used to create a linear model of the vehicle, updated at preset increments throughout a nominal, no-wind flight. This linear representation of the vehicle pitch dynamics remained valid in the face of small variations in pitch or angle-of-attack that might exist throughout any given ascent trajectory due to wind variations. This data was saved in the form of the linear, time-invariant (LTI), discrete A, B, C and D state-space matrices shown in Equation 3.1.

[This page intentionally left blank.]

Chapter 3

MPC Controller Theory

3.1 MPC Overview

Model Predictive Control, as a general concept, has existed for many years. Because of MPC's large computational requirements, widespread interest first began during the 1980's in applications with low-rate control requirements and low-bandwidths, such as those processes found in the chemical and process control industries [2]. Only recently, with significant increases in computer processor speeds, has it become feasible to apply MPC to high bandwidth systems such as flight vehicles, which require high-rate control processes.
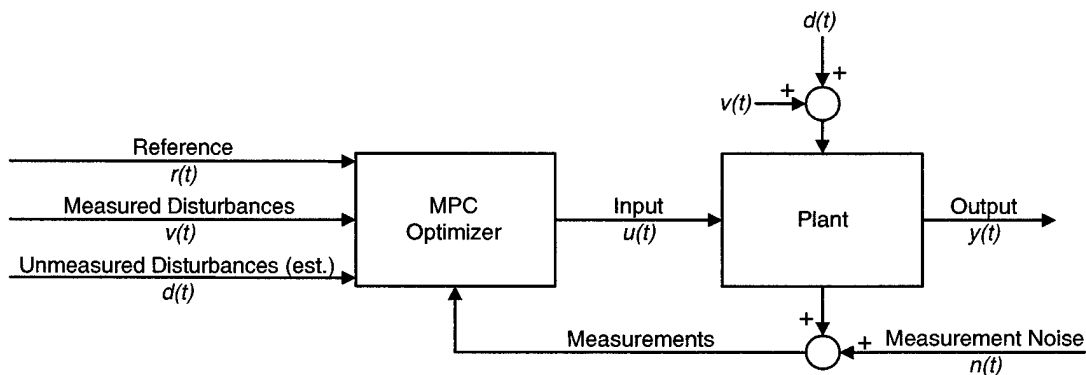
Figure 3-1: MPC Operation Graphical Overview
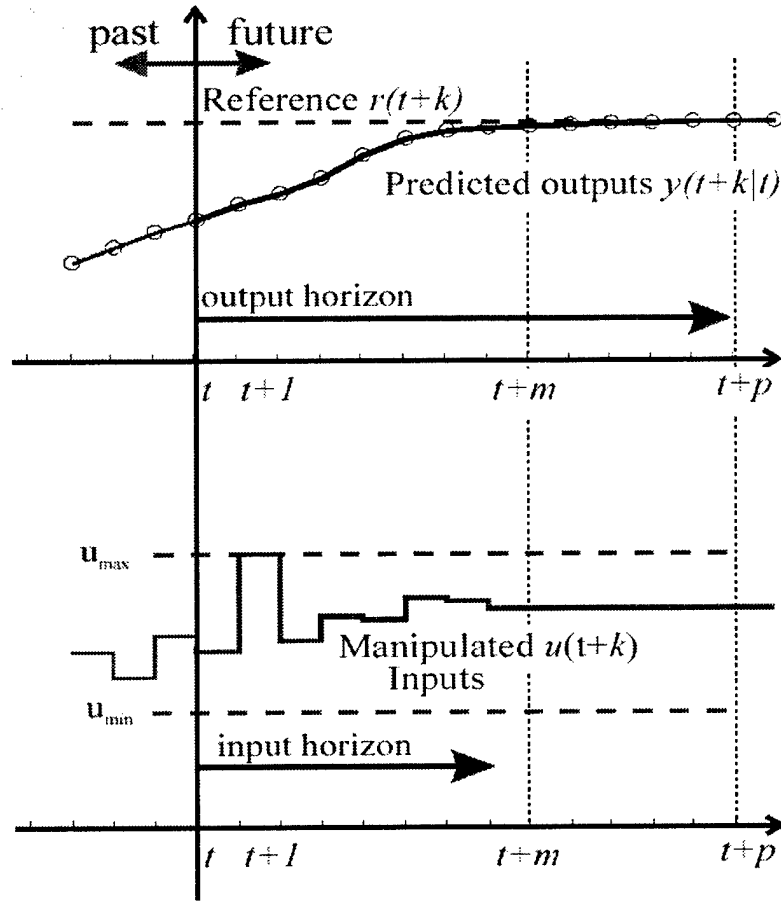
Figure 3-2: MPC Operation Graphical Overview

The conceptual structure of MPC is shown in Figure 3-1. As its name implies, MPC uses an internal model and knowledge of future and present measured and estimated unmeasured disturbances, reference trajectories and current states to predict the output of a given system over a finite time horizon, known as the prediction, or output horizon, p. Recent wind studies have shown that wind behavior does not change in any statistically significant way over relatively short times, such as five minutes or less. Since current LIDAR wind sensors are capable of measuring the wind field in less than five minutes, they can provide "future" disturbance information to a MPC controller by simply providing the current wind information at the location where the vehicle will be at the appropriate times in the future. Because this research utilizes full disturbance information provided by the LIDAR wind sensor,

all unmeasured disturbance estimates are neglected. However, for completeness sake, the unmeasured disturbance terms are included in the MPC matrix derivations. A graphical description of this process is shown in Figure 3-2. This prediction ability allows MPC to solve an optimal control problem on-line in order to minimize the error between the predicted output and a desired output trajectory, also known as the reference trajectory. This process could be subject to constraints on the manipulated inputs and outputs. The result of the optimization is a series of optimal input commands, each applying to a given time step within a second horizon, called the control or input horizon, m, also shown in Figure 3-2. Between the control and prediction horizon, the input command is held constant. At each time step, only the first input command is applied to the system. The remainder of the optimal input commands are discarded and the system is propagated forward one time step. At this point the optimal control problem is solved again, with an appropriate update to all measurements from the plant, as well as the applicable disturbance and reference trajectories. As this process is repeated and the system progresses through time, the projection and control horizons recede at the same rate.

3.2   Unconstrained MPC Optimal Solution

Consider the discrete LTI plant with the state and output equations as follows:

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \qquad (3.1)$$

$$y(k) = Cx(k) + D_v v(k) + D_d d(k)$$

where $x(k) \in \Re^{nx \times 1}$, $u(k) \in \Re^{nu \times 1}$, $y(k) \in \Re^{ny \times 1}$, $v(k) \in \Re^{nv \times 1}$, and $d(k) \in \Re^{nd \times 1}$

As stated previously, $v(k)$ and $d(k)$ represent the measured and estimated unmeasured disturbances experienced by the plant, respectively. In the same way, $u(k)$ and $y(k)$ represent the input to and output from the plant model and $x(k)$ represents the present model states. Model Predictive Control minimizes the cost function shown in

Equation 3.5 to find the optimal input command series, in the presence of the three error terms shown in that equation and their respective weighting matrices, w_u, w_Δ and w_y. These equations are shown below.

$$\sum_{i=0}^{p-1} [u(k+i) - u_T(k+i)]^2 w_u(i) \tag{3.2}$$

$$\sum_{i=0}^{m-1} z(k+i)^2 w_\Delta(i) \tag{3.3}$$

$$\sum_{i=0}^{p-1} [y(k+i) - r(k+i)]^2 w_y(i) \tag{3.4}$$

This cost function penalizes the error between the commanded output and the predicted output (Equation 3.4), the trim input and the predicted input (Equation 3.2), and the difference between the current input command and the previous input command (Equation 3.3) as shown by Equation 3.6. Thus, it exhibits features of cost functions found in both [2] and [4], regulating both inputs and differences between consecutive inputs. $\bar{z}$ is called the optimization parameter. Because the plant in this research only has one output, an overline represents a variable which is in vector form. As shown in Equation 3.5, the optimal $\bar{z}(k)$ must be found to minimize the cost function, J. The optimal $\bar{z}(k)$ will be denoted as $\bar{z}^*(k)$.

$$J(\bar{z}(k)) = \min_{\bar{z}(k) \in \Re^{(m*nu) \times 1}} \sum_{i=0}^{p-1} [u(k+i) - u_T(k+i)]^2 w_u(i) + \sum_{i=0}^{m-1} z(k+i)^2 w_\Delta(i) +$$
$$\sum_{i=0}^{p-1} [y(k+i) - r(k+i)]^2 w_y(i) \tag{3.5}$$

Several other parameters must be defined before continuing. These include nu and ny, which represent the number of inputs to and outputs from the plant, respectively. nd represents the number of estimated unmeasured disturbances and nv represents the number of measured disturbances. Finally, nx represents the number of model

states.

The optimization parameter $\overline{z}(k)$ is related to the variation of the input variables as shown below:

$$\overline{\Delta u}(k) = J_M \overline{z}(k) \quad \text{where} \quad J_M \in \Re^{(p*nu) \times (m*nu)} \tag{3.6}$$

$$\overline{\Delta u}(k) = \overline{u}(k) - \overline{u}(k-1)$$

J_M is a matrix used to impose additional constraints on the optimum $\overline{\Delta u}(k)$. The function and structure of this matrix are discussed in Section 3.4, Blocking and Basis Functions.

To implement this cost function in an effective and flexible manner, it should be rewritten in matrix form. To do this, several new terms must be defined, as shown below.

$$\overline{u}(k) = I_p u(k-1) + K_1 \overline{\Delta u}(k) \tag{3.7}$$

$$I_1 = Identity(nu \times nu) \tag{3.8}$$

$$I_p = \begin{bmatrix} I_1 \\ I_1 \\ \vdots \\ I_1 \end{bmatrix} \in \Re^{(p*nu) \times nu} \tag{3.9}$$

$$K_1 = \begin{bmatrix} I_1 & 0 & \cdots & 0 \\ I_1 & I_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_1 & I_1 & \cdots & I_1 \end{bmatrix} \in \Re^{(p*nu) \times (p*nu)} \tag{3.10}$$

We can now write this cost function in matrix form:

41

$$J = \min_{\bar{z}(k) \in \Re^{(m*nu) \times 1}} [\bar{u}(k) - \bar{u}_T(k)]^T W_u [\bar{u}(k) - \bar{u}_T(k)] + \bar{z}^T(k) J_M^T W_\Delta J_M \bar{z}(k) +$$

$$[\bar{y}(k) - \bar{r}(k)]^T W_y [\bar{y}(k) - \bar{r}(k)] \quad (3.11)$$

If no constraints exist, a closed-form analytical minimum of the cost function can be found. This process is described in the following section and was derived from an initial treatment in [1].

Let $y(k+i|k)$ and $\overline{\Delta u}(k+i|k)$ be the output and input change predictions obtained by iterating the model i times into the future, from the current state, k. As previously defined, p and m are the prediction and control horizons, respectively. The vectors $\bar{y}(k)$ and $\overline{\Delta u}(k)$ are structured as shown below.

$$\bar{y}(k) = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+p|k) \end{bmatrix} \in \Re^{(p*ny) \times 1} \quad (3.12)$$

$$\overline{\Delta u}(k) = \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+p-1|k) \end{bmatrix} \in \Re^{(p*nu) \times 1} \quad (3.13)$$

Using this convention, we can also write the prediction of the input in vector form, as shown below.

$$\bar{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+p-1) \end{bmatrix} \in \Re^{(p*nu) \times 1} \quad (3.14)$$

The known input, measured disturbance and estimated unmeasured disturbance

trajectories are now defined as shown below.

$$\overline{u_T}(k) = \begin{bmatrix} u_T(k) \\ u_T(k+1) \\ \vdots \\ u_T(k+p-1) \end{bmatrix} \in \Re^{(p*nu)\times 1} \tag{3.15}$$

$$\overline{v}(k) = \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+p) \end{bmatrix} \in \Re^{(p*nu+nv)\times 1} \tag{3.16}$$

$$\overline{d}(k) = \begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+p) \end{bmatrix} \in \Re^{(p*nu+nd)\times 1} \tag{3.17}$$

At this point we are ready to begin calculating $\overline{y}(k)$. We will begin with the standard discrete LTI state-space equations, as shown in Equation 3.1. This equation shows the system at time step k.

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \tag{3.18}$$
$$y(k) = Cx(k) + D_v v(k) + D_d d(k)$$

The system equations at time step $k+1$ are written below.

$$x(k+2) = Ax(k+1) + B_u u(k+1) + B_v v(k+1) + B_d d(k+1) \tag{3.19}$$
$$y(k+1) = Cx(k+1) + D_v v(k+1) + D_d d(k+1)$$

To reach the complete realization of these equations, the terms of $x(k+1)$ from

Equation 3.18, must be inserted where appropriate into Equation 3.19, as shown below.

$$x(k+2) = A[Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)] +$$
$$B_u u(k+1) + B_v v(k+1) + B_d d(k+1)$$
(3.20)

$$y(k+1) = C[Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)] +$$
$$D_v v(k+1) + D_d d(k+1)$$

Writing the system equations yet again, for time step $k+2$, gives the following equations.

$$x(k+3) = Ax(k+2) + B_u u(k+2) + B_v v(k+2) + B_d d(k+2)$$
(3.21)
$$y(k+2) = Cx(k+2) + D_v v(k+2) + D_d d(k+2)$$

As before, the terms of $x(k+2)$ must be inserted where appropriate into Equation 3.21, as shown below.

$$x(k+3) = A[A[Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)] +$$
$$B_u u(k+1) + B_v v(k+1) + B_d d(k+1)] +$$
(3.22)
$$B_u u(k+2) + B_v v(k+2) + B_d d(k+2)$$

$$y(k+2) = C[A[Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)] +$$
$$B_u u(k+1) + B_v v(k+1) + B_d d(k+1)] +$$
$$D_v v(k+2) + D_d d(k+2)$$

Noticing the emerging pattern, the prediction of $\bar{y}$, at time k, to any time $k+i$, can be expressed as:

$$y(k+i|k) =$$

$$C \left[A^i x(k) + \sum_{h=0}^{i-1} A^{i-1-h} B_u \left[u(k-1) + \sum_{j=0}^{h} \Delta u(k+j) + B_v v(k+h) + B_d d(k+h) \right] \right]$$

$$+ D_v v(k+i) + D_d d(k+i) \quad (3.23)$$

To realize this equation in vector form, several additional matrices must be defined, as shown below.

$$\overline{y}(k) = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+p|k) \end{bmatrix} \in \Re^{(p*ny) \times 1} \quad (3.24)$$

$$S_x = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \in \Re^{(p*ny) \times nx} \quad (3.25)$$

$$S_{u1} = \begin{bmatrix} CB_u \\ CB_u + CAB_u \\ \vdots \\ \sum_{h=0}^{p-1} CA^h B_u \end{bmatrix} \in \Re^{(p*ny) \times nu} \quad (3.26)$$

$$S_u = \begin{bmatrix} CB_u & 0 & 0 & 0 \\ CB_u + CAB_u & CB_u & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \sum_{h=0}^{p-1} CA^h B_u & \sum_{h=0}^{p-2} CA^h B_u & \cdots & CB_u \end{bmatrix} \in \Re^{(p*ny) \times (p*nu)} \quad (3.27)$$

$$H_v = \begin{bmatrix} CB_v & D_v & 0 & \cdots & 0 \\ CAB_v & CB_v & D_v & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B_v & CA^{p-2}B_v & CA^{p-3}B_v & \cdots & D_v \end{bmatrix} \in \Re^{(p*ny) \times (p*nu+nv)} \quad (3.28)$$

$$H_d = \begin{bmatrix} CB_d & D_d & 0 & \cdots & 0 \\ CAB_d & CB_d & D_d & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B_d & CA^{p-2}B_d & CA^{p-3}B_d & \cdots & D_d \end{bmatrix} \in \Re^{(p*ny) \times (p*nu+nd)} \quad (3.29)$$

Using these matrices, the prediction of $\overline{y}(k)$ can now be rewritten as shown below:

$$\overline{y}(k) = S_x x(k) + S_{u1} u(k-1) + S_u \overline{\Delta u}(k) + H_v \overline{v}(k) + H_d \overline{d}(k) \quad (3.30)$$

Likewise, the cost function in its most basic matrix form can now be written as:

$$J = [\overline{u}(k) - \overline{u_T}(k)]^T W_u [\overline{u}(k) - \overline{u_T}(k)] + \overline{\Delta u}^T(k) W_\Delta \overline{\Delta u}(k) +$$
$$[\overline{y}(k) - \overline{r}(k)]^T W_y [\overline{y}(k) - \overline{r}(k)] \quad (3.31)$$

To find the analytical minimum of this cost function, the derivative of the function must be taken. The cost function will be broken into three separate parts, J_1, J_2 and J_3 for simplicity. In addition, the (k) convention will be neglected for brevity's sake. Taking the first term of Equation 3.31 to be J_1 we have:

$$J_1 = [\overline{u} - \overline{u_T}]^T W_u [\overline{u} - \overline{u_T}] \quad (3.32)$$

Substituting from Equation 3.7 yields:

$$J_1 = [I_p u(-1) + K_1 \overline{\Delta u} - \overline{u}_T]^T W_u [I_p u(-1) + K_1 \overline{\Delta u} - \overline{u}_T] \qquad (3.33)$$

Substituting once again from Equation 3.6 yields:

$$J_1 = [I_p u(-1) + K_1 J_M \overline{z} - \overline{u}_T]^T W_u [I_p u(-1) + K_1 J_M \overline{z} - \overline{u}_T] \qquad (3.34)$$

Taking the partial now yields:

$$\frac{\partial J_1}{\partial \overline{z}} = 2[I_p u(-1) - \overline{u}_T]^T W_u K_1 J_M + 2\overline{z}^T J_M^T K_1^T W_u K_1 J_M \qquad (3.35)$$

Taking the second term of Equation 3.31 as J_2 and making the appropriate substitution from Equation 3.6, we have:

$$J_2 = \overline{\Delta u}^T W_\Delta \overline{\Delta u} = \overline{z}^T J_M^T W_\Delta J_M \overline{z} \qquad (3.36)$$

Taking the partial yields:

$$\frac{\partial J_2}{\partial \overline{z}} = 2\overline{z}^T J_M^T W_\Delta J_M \qquad (3.37)$$

Finally, taking the third term of Equation 3.31 as J_3 we have:

$$J_3 = [\overline{y} - \overline{r}]^T W_y [\overline{y} - \overline{r}] \qquad (3.38)$$

Inserting Equation 3.30 where appropriate yields:

$$J_3 = [S_x x + S_{u1} u(-1) + S_u \overline{\Delta u} + H_v \overline{v} + H_d \overline{d} - \overline{r}]^T W_y$$
$$[S_x x + S_{u1} u(-1) + S_u \overline{\Delta u} + H_v \overline{v} + H_d \overline{d} - \overline{r}] \quad (3.39)$$

Combining all constants, we can define a new term F as:

$$F = [S_x x + S_{u1} u(-1) + H_v \overline{v} + H_d \overline{d} - \overline{r}] \qquad (3.40)$$

47

Inserting F, J_3 is now:

$$J_3 = [F + S_u\overline{\Delta u}]^T W_y [F + S_u\overline{\Delta u}] \tag{3.41}$$

Taking the partial yields:

$$\frac{\partial J_3}{\partial \overline{z}} = 2F^T W_y S_u J_M + 2\overline{z}^T J_M^T S_u^T W_y S_u J_M \tag{3.42}$$

Setting the derivatives equal to zero and solving will yield the function minimum.

$$\frac{\partial J}{\partial \overline{z}} = \frac{\partial J_1}{\partial \overline{z}} + \frac{\partial J_2}{\partial \overline{z}} + \frac{\partial J_3}{\partial \overline{z}} = 0 \tag{3.43}$$

From above, this is equal to:

$$[I_p u(-1) - \overline{u}_T]^T W_u K_1 J_M + \overline{z}^{*T} J_M^T K_1^T W_u K_1 J_M + \overline{z}^{*T} J_M^T W_\Delta J_M +$$
$$F^T W_y S_u J_M + \overline{z}^{*T} J_M^T S_u^T W_y S_u J_M = 0 \tag{3.44}$$

Consolidating the $\overline{z}^*$ terms yields:

$$\overline{z}^{*T}(J_M^T K_1^T W_u K_1 J_M + J_M^T W_\Delta J_M + J_M^T S_u^T W_y S_u J_M) =$$
$$- F^T W_y S_u J_M - [I_p u(-1) - \overline{u}_T]^T W_u K_1 J_M \tag{3.45}$$

A new matrix, K_{du}, can be defined as:

$$K_{du} = (J_M^T K_1^T W_u K_1 J_M + J_M^T W_\Delta J_M + J_M^T S_u^T W_y S_u J_M) \tag{3.46}$$

Equation 3.45 can now be rewritten as:

$$\overline{z}^* = -K_{du}^{-1}[F^T W_y S_u J_M - [I_p u(-1) - \overline{u}_T]^T W_u K_1 J_M]^T \tag{3.47}$$

Reinserting the terms of F yields:

48

$$\overline{z}^* = -K_{du}^{-1}[(S_x x + S_{u1} u(-1) + H_v \overline{v} + H_d \overline{d} - \overline{r})^T W_y S_u J_M -$$

$$[I_p u(-1) - \overline{u}_T]^T W_u K_1 J_M]^T \quad (3.48)$$

One final reorganization yields the equation shown below.

$$\overline{z}^* = -K_{du}^{-1}[-\overline{r}^T W_y S_u J_M + (H_v \overline{v} + H_d \overline{d})^T W_y S_u J_M +$$

$$u^T(-1)(I_p^T W_u K_1 J_M + S_{u1}^T W_y S_u J_M) - \overline{u}_T^T W_u K_1 J_M + x^T S_x^T W_y S_u J_M]^T \quad (3.49)$$

This equation can now be separated into the following matrices.

$$K_r = [-W_y S_u J_M] \in \Re^{(p*ny)\times(m*nu)} \quad (3.50)$$

$$K_v = [H_v^T W_y S_u J_M] \in \Re^{(p*nu+nv)\times(m*nu)} \quad (3.51)$$

$$K_d = [H_d^T W_y S_u J_M] \in \Re^{(p*nu+nd)\times(m*nu)} \quad (3.52)$$

$$K_u = [I_p^T W_u K_1 J_M + S_{u1}^T W_y S_u J_M] \in \Re^{nu\times(m*nu)} \quad (3.53)$$

$$K_T = [-W_u K_1 J_M] \in \Re^{(p*nu)\times(m*nu)} \quad (3.54)$$

$$K_x = [S_x^T W_y S_u J_M] \in \Re^{nx\times(m*nu)} \quad (3.55)$$

$$K_{du} = [J_M^T K_1^T W_u K_1 J_M + J_M^T W_\Delta J_M + J_M^T S_u^T W_y S_u J_M] \in \Re^{(m*nu)\times(m*nu)} \quad (3.56)$$

Using these matrices, $\overline{z}^*$ can be defined as:

$$\overline{z}^*(k) = -K_{du}^{-1}[\overline{r}^T(k)K_r + \overline{v}^T(k)K_v + \overline{d}^T(k)K_d +$$

$$u^T(k-1)K_u + \overline{u}_T^T(k)K_T + x^T(k)K_x]^T \quad (3.57)$$

One final substitution from Equation 3.6 is required to reach the optimal input vector $\overline{\Delta u}^*$, the first term of which will be implemented as the system input for the

current time step.

$$\overline{\Delta u}^* = J_M \overline{z}^*$$ (3.58)

3.3 Constrained MPC Optimal Solution

This cost function can also be minimized subject to linear equality constraints in the inputs, input-variations and outputs. A synopsis of these constraints are shown below.

$$u_i^{min} \leq u(k+i|k) \leq u_i^{max}$$ (3.59)

$$\Delta u_i^{min} \leq \Delta u(k+i|k) \leq \Delta u_i^{max}, \quad i = 0, \cdots, p-1$$ (3.60)

$$-\epsilon + y_i^{min} \leq y(k+i+1|k) \leq y_i^{max} + \epsilon$$ (3.61)

$$\Delta u(k+j|k) = 0, \quad j = m, \cdots, p$$ (3.62)

$$\epsilon \leq 0$$ (3.63)

Constraints on the inputs and input-variations are treated as *hard* constraints, meaning that they are never violated. These constraints would be equivalent to the physical limits of a given actuator. To avoid infeasible optimization problems, the output constraints are treated at *soft* constraints, meaning that they may be violated, if required to maintain optimization feasibility. This is allowed by an addition to the cost function called a slack variable, ϵ. The slack variable is normally equal to zero. However, when a given output crosses it's constraint limit, ϵ becomes equal to the magnitude of violation. ϵ is then squared, multiplied by a weighting factor, ρ_ϵ, and added to the function cost, which was shown in Equation 3.5. Thus, while violating constraints greatly increases the cost of a given input sequence, the optimization problem will never become infeasible.

Using the previously defined matrices, we can now write the cost function to be optimized, including the slack variable term:

$$J = \rho_\epsilon \epsilon^2 + z^T K_{du} z + 2[\bar{r}^T K_r + \bar{v}^T K_v + u^T(k-1)K_u + u_T^T K_T + x^T(k)K_x]z \quad (3.64)$$

Likewise, the constraints can be realized as:

$$
\begin{bmatrix}
y^{min}(k+1) \\
\vdots \\
y^{min}(k+p) \\
u^{min}(k) \\
\vdots \\
u^{min}(k+p-1) \\
\Delta u^{min}(k) \\
\vdots \\
\Delta u^{min}(k+p-1)
\end{bmatrix}
\leq
\begin{bmatrix}
y(k+1) \\
\vdots \\
y(k+p) \\
u(k) \\
\vdots \\
u(k+p-1) \\
\Delta u(k) \\
\vdots \\
\Delta u(k+p-1)
\end{bmatrix}
\leq
\begin{bmatrix}
y^{max}(k+1) \\
\vdots \\
y^{max}(k+p) \\
u^{max}(k) \\
\vdots \\
u^{max}(k+p-1) \\
\Delta u^{max}(k) \\
\vdots \\
\Delta u^{max}(k+p-1)
\end{bmatrix}
\quad (3.65)
$$

Equation 3.64 can now be minimized, subject to Equation 3.65 and the constraint $\epsilon \geq 0$. This optimization can be solved using the optimizer of choice. An excellent implementation of this constrained optimal control can be found in a controller created by Dr. Alberto Bemporad and associates from the Automatic Control Laboratory in Zurich, Switzerland. This controller uses the Matlab optimizer routine, DANTZGMP.m to solve the quadratic problem previously shown. For a slightly more detailed treatment of the constrained equation setup, see [1]. Because of the limited time and computational resources available, this research will deal exclusively with unconstrained MPC control theory.

3.4 Blocking and Basis Functions

As mentioned previously, J_M is a matrix used to impose additional constraints on the optimum $\overline{\Delta u}(k)$. This matrix can impose either a blocking function or a basis

function on the optimum solution. The matrix J_M decreases the complexity of the optimization problem while still allowing, in most cases, enough freedom for the optimizer to solve the problem. Utilizing the matrix J_M can significantly reduce the number of rows in the matrices shown in Equation 3.57, thus lowering the calculation load for every optimizer cycle.

Blocking Functions

To further constrain the optimization problem, J_M can impose blocking requirements on the problem. This means that the terms in $\overline{z}$ act *individually* to constrain the terms of $\overline{\Delta u}$ at certain times during the projection horizon. To illustrate this concept, an example is offered. Let the matrix J_M be defined as shown below. In all cases, the variable m is equal to the number of columns in J_M. This would require that $u(0) = u(1)$, $u(2) = u(3) = u(4)$ and $u(5) = u(6)$.

$$
J_M = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \quad \text{dictates that} \quad \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_4 \\ \Delta u_5 \\ \Delta u_6 \end{bmatrix} = \begin{bmatrix} z_1 \\ 0 \\ z_2 \\ 0 \\ 0 \\ z_3 \\ 0 \end{bmatrix}
$$

Figure 3-3 depicts a possible input and input variation schedule which could result from this example J_M.
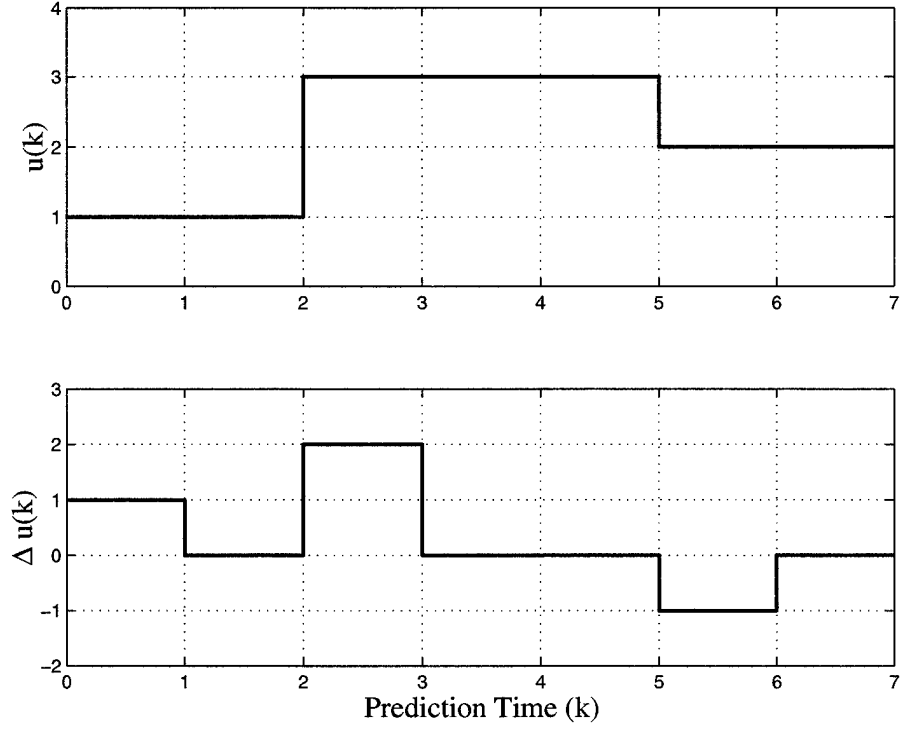
Figure 3-3: 1st Blocking function example: Inputs and Input Increments

A second example, showing another variation of blocking functions is included below. Again, the J_M representing this function is shown, as well as the input variation schedule which it would produce. This J_M matrix would required that $\Delta u(0) = \Delta u(1)$, $\Delta u(2) = \Delta u(3) = \Delta u(4)$ and $\Delta u(5) = \Delta u(6)$.

$$
J_M = \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & I & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix} \quad \text{dictates that} \quad \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_4 \\ \Delta u_5 \\ \Delta u_6 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_1 \\ z_2 \\ z_2 \\ z_2 \\ z_3 \\ z_3 \end{bmatrix}
$$

Figure 3-4 depicts a possible input and input variation schedule which could result from this blocking function.
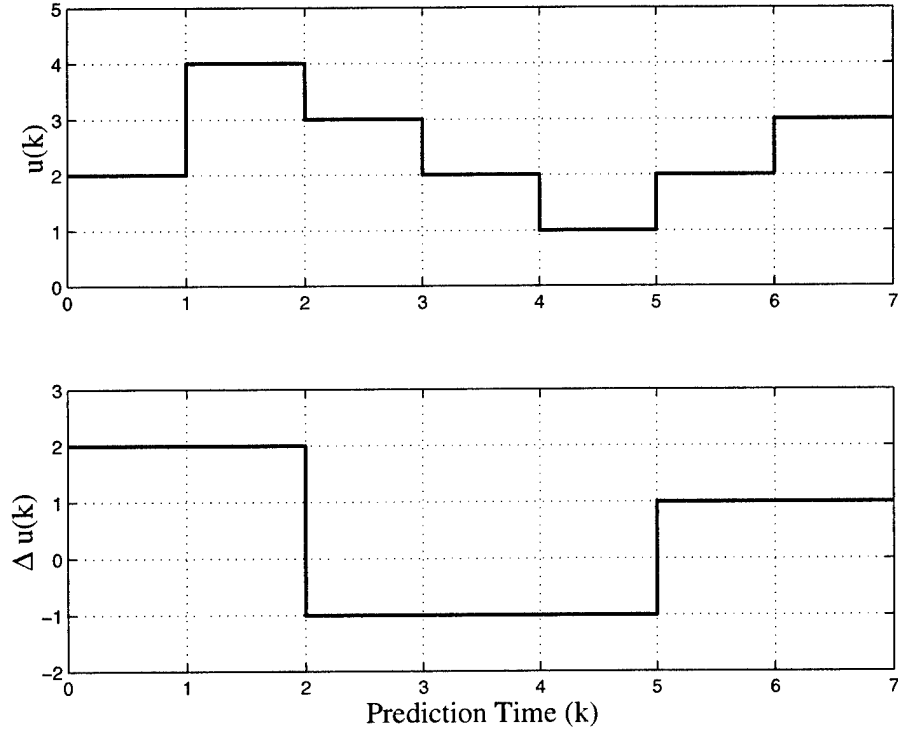
Figure 3-4: 2nd Blocking function example: Inputs and Input Increments

Basis Functions

Another approach to constraining the optimization problem is to use the matrix J_M as a basis function. This means that the terms in $\bar{z}$ act *in concert* to constrain the terms of $\overline{\Delta u}$ during the projection horizon. To illustrate this concept, an example is offered. Let the matrix J_M be defined as shown on page 55. In this case, each column is a separate basis function. The terms of $\bar{z}$ determine the weighting placed on each function in the resulting $\overline{\Delta u}$, which is also shown. For simplicity's sake, we will assume that the number of inputs, nu, is one.

$$J_M = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0 \\ 1 & 1 & 0.33 \\ 1 & 1 & 0.66 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{dictates that} \quad \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_4 \\ \Delta u_5 \\ \Delta u_6 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_1 \\ z_1 + 0.5z_2 \\ z_1 + z_2 + 0.33z_3 \\ z_1 + z_2 + 0.66z_3 \\ z_1 + z_2 + z_3 \\ z_1 + z_2 + z_3 \end{bmatrix}$$

Because of the integration of the matrix J_M into the cost function, as previously shown, J_M reduces the size of the problem linearly from size p to size m, without sacrificing significant accuracy. The results of a comparison between the performance of a system using the J_M matrix shown above and that of a system using an identity J_M matrix of size p are shown below. An identity J_M matrix represents a system with no imposed blocking or basis function constraints.

	Example J_M	Identity J_M
Process Time (sec)	0.0200	0.0500
Function Cost (-)	0.0364	0.0321

Table 3.1: J_M performance comparison

Utilizing the example J_M yields an increase in the cost function J of approximately 13.4%. However, it also decreases the processing time necessary to complete one computation cycle by 60%. This trade between minimizing the cost and lowering the processing speed exists for all J_M matrices, with the most optimum solution, in terms of cost, being that offered by the identity J_M matrix. The optimum solution, in terms of speed, is dependant upon the control task requirements.

3.5 MPC Example

To further clarify the process used to calculate an input command using MPC, an example is offered. The example discrete state space system will be drawn from the K-1 linearized model at the point of maximum dynamic pressure, approximately 80 seconds into a nominal flight. The matrices and necessary trajectories are shown below. As previously stated, no estimated unmeasured disturbances, $\bar{d}(k)$, were considered for this research. The previously discussed example J_M will be used, indicating a prediction horizon of 7.

As shown below, there are four states, $[\dot{\theta} \; \theta \; u \; w]$, where u and w are the body velocities in the X and Z inertial directions respectively, as defined in Chapter 2 and θ is vehicle pitch. The input is the pitch gimbal command, δ_p and the disturbance is the wind velocity in the negative Z direction. The output is the angle of attack (α) of the body.

$$
A = \begin{bmatrix} 0 & 0.4504 & -0.0001 & 0.0003 \\ 1 & 0 & 0 & 0 \\ 0 & 33.1546 & -0.0106 & 0.0042 \\ 0 & -75.5361 & 0.0047 & -0.0192 \end{bmatrix} \quad B_u = \begin{bmatrix} 3.1761 \\ 0 \\ -25.5243 \\ 57.0848 \end{bmatrix} \quad B_v = \begin{bmatrix} -0.0003 \\ 0 \\ -0.0042 \\ 0.0192 \end{bmatrix}
$$

$$
C = \begin{bmatrix} 0 & 1 & -0.0003 & 0.0007 \end{bmatrix} \quad D_v = \begin{bmatrix} -0.0007 \end{bmatrix}
$$

In addition, the input weighting matrices, W_u and W_Δ, are both identity matrices of size p, while the output weighting matrix, W_y, is two times an identity matrix of size p.

Next we will define the reference trajectory, $\bar{r}(k)$, the disturbance trajectory, $\bar{v}(k)$, the nominal, or trim, output trajectory, $\bar{y_0}(k)$, the nominal input trajectory, $\bar{u_0}(k)$, the nominal state, $x_0(k)$, the actual state, $x(k)$ and the most recent input, $u(k-1)$. $\bar{r}(k)$ represents the commanded α trajectory. $\bar{v}(k)$ represents the wind disturbance which is projected to affect the plant at each time step. Notice that this vector is one term

longer than all other trajectory vectors, due to the prediction formulation. $\overline{y_0}(k)$, $\overline{u_0}(k)$ and $x_0(k)$ represent the trim output, input and state values that were used during the linearization process. $u(k-1)$ represents the gimbal command calculated and implemented during the previous optimization processor cycle.

$$
\overline{r}(k) = \begin{bmatrix} -0.0222 \\ -0.0210 \\ -0.0199 \\ -0.0187 \\ -0.0175 \\ -0.0164 \\ -0.0146 \end{bmatrix}
\qquad
\overline{v}(k) = \begin{bmatrix} 151 \\ 149 \\ 147 \\ 144 \\ 151 \\ 158 \\ 155 \\ 157 \end{bmatrix}
\qquad
\overline{y_0}(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\qquad
\overline{u_0}(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
x_0(k) = \begin{bmatrix} 0 \\ -0.4204 \\ 1120 \\ 500.6 \end{bmatrix}
\qquad
x(k) = \begin{bmatrix} -0.0229 \\ -0.3188 \\ 1126 \\ 549.8 \end{bmatrix}
\qquad
u(k-1) = \begin{bmatrix} -0.01474 \end{bmatrix}
$$

Using these variables, the following matrices and vectors were created, using the previously defined procedures.

$$
I_p = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\qquad
K_1 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

$$S_x = \begin{bmatrix} 1.0000 & -0.0628 & 0.0000 & -0.0000 \\ -0.0628 & 0.4517 & -0.0001 & 0.0003 \\ 0.4517 & -0.0543 & 0.0000 & -0.0000 \\ -0.0543 & 0.2056 & -0.0000 & 0.0001 \\ 0.2056 & -0.0362 & 0.0000 & -0.0000 \\ -0.0362 & 0.0943 & -0.0000 & 0.0001 \\ 0.0943 & -0.0217 & 0.0000 & -0.0000 \end{bmatrix} \qquad S_{u1} = \begin{bmatrix} 0.0476 \\ 3.2227 \\ 3.0429 \\ 4.4760 \\ 4.3124 \\ 4.9643 \\ 4.8533 \end{bmatrix}$$

$$S_u = \begin{bmatrix} 0.0476 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3.2227 & 0.0476 & 0 & 0 & 0 & 0 & 0 \\ 3.0429 & 3.2227 & 0.0476 & 0 & 0 & 0 & 0 \\ 4.4760 & 3.0429 & 3.2227 & 0.0476 & 0 & 0 & 0 \\ 4.3124 & 4.4760 & 3.0429 & 3.2227 & 0.0476 & 0 & 0 \\ 4.9643 & 4.3124 & 4.4760 & 3.0429 & 3.2227 & 0.0476 & 0 \\ 4.8533 & 4.9643 & 4.3124 & 4.4760 & 3.0429 & 3.2227 & 0.0476 \end{bmatrix}$$

$$H_v = \begin{bmatrix} 0.0147 & -0.7000 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.3003 & 0.0147 & -0.7000 & 0 & 0 & 0 & 0 & 0 \\ 0.0250 & -0.3003 & 0.0147 & -0.7000 & 0 & 0 & 0 & 0 \\ -0.1360 & 0.0250 & -0.3003 & 0.0147 & -0.7000 & 0 & 0 & 0 \\ 0.0191 & -0.1360 & 0.0250 & -0.3003 & 0.0147 & -0.7000 & 0 & 0 \\ -0.0621 & 0.0191 & -0.1360 & 0.0250 & -0.3003 & 0.0147 & -0.7000 & 0 \\ 0.0122 & -0.0621 & 0.0191 & -0.1360 & 0.0250 & -0.3003 & 0.0147 & -0.7000 \end{bmatrix} *1e{-}3$$

$$K_r = \begin{bmatrix} -0.0952 & 0 & 0 \\ -6.5407 & 0 & 0 \\ -12.6264 & -0.0476 & 0 \\ -21.5783 & -3.3179 & -0.0314 \\ -30.2032 & -9.5835 & -2.1898 \\ -40.1318 & -17.1024 & -6.3575 \\ -49.8383 & -25.8908 & -13.5114 \end{bmatrix} \qquad K_v = \begin{bmatrix} -0.0059 & -0.0010 & -0.0002 \\ -0.0097 & -0.0025 & -0.0010 \\ -0.0146 & -0.0026 & -0.0006 \\ -0.0234 & -0.0060 & -0.0023 \\ -0.0255 & -0.0067 & -0.0016 \\ -0.0355 & -0.0142 & -0.0055 \\ -0.0274 & -0.0116 & -0.0043 \\ -0.0349 & -0.0181 & -0.0095 \end{bmatrix}$$

$$K_u = \begin{bmatrix} 755.4407 & 279.3801 & 113.0192 \end{bmatrix} \qquad K_T = \begin{bmatrix} -1.0000 & 0 & 0 \\ -2.0000 & 0 & 0 \\ -3.0000 & -0.5000 & 0 \\ -4.0000 & -1.5000 & -0.3300 \\ -5.0000 & -2.5000 & -0.9900 \\ -6.0000 & -3.5000 & -1.9900 \\ -7.0000 & -4.5000 & -2.9900 \end{bmatrix}$$

$$K_x = \begin{bmatrix} 13.6717 & 3.6333 & 1.4921 \\ 8.3081 & 1.3831 & 0.2333 \\ -0.0020 & -0.0003 & -0.0001 \\ 0.0059 & 0.0010 & 0.0002 \end{bmatrix} \qquad K_{du} = \begin{bmatrix} 2.9842 & 1.2462 & 0.5398 \\ 1.2462 & 0.5783 & 0.2662 \\ 0.5398 & 0.2662 & 0.1304 \end{bmatrix} * 1e3$$

These structures were used to calculate $\bar{z}^*(k)$ (Equation 3.57), which was used to calculate $\overline{\Delta u}^*(k)$ (Equation 3.58), which was used to calculate $\bar{y}(k)$ (Equation 3.30) and $\bar{u}^*(k)$ (Equation 3.7), which were used to calculate the function cost, J (Equation 3.31). These vectors are shown on page 60.

$$
\overline{z}^*(k) = \begin{bmatrix} 0.0261 \\ -0.0442 \\ 0.0217 \end{bmatrix}
\qquad
\overline{\Delta u}^*(k) = \begin{bmatrix} 0.0261 \\ 0.0261 \\ 0.0040 \\ -0.0109 \\ -0.0037 \\ 0.0037 \\ 0.0037 \end{bmatrix}
\qquad
\overline{y}(k) = \begin{bmatrix} -0.1315 \\ -0.0466 \\ -0.0376 \\ 0.0070 \\ -0.0315 \\ -0.0188 \\ -0.0262 \end{bmatrix}
$$

$$
\overline{u}^*(k) = \begin{bmatrix} 0.0114 \\ 0.0375 \\ 0.0416 \\ 0.0307 \\ 0.0269 \\ 0.0306 \\ 0.0342 \end{bmatrix}
\qquad
J = \begin{bmatrix} 0.0364 \end{bmatrix}
$$

Chapter 4

MPC Architecture Development

4.1   Architecture Candidates

Many possible control layout scenarios exist for inserting an MPC controller into the outer and possibly inner loops of a rocket control system. In order to decide which MPC control architectures should be evaluated in this thesis, a detailed linear evaluation was conducted on five possible control layout candidates. Based on the results of this analysis, two of the architectures were chosen for further study. The five candidate architectures are shown below and on the following pages, along with short descriptions.
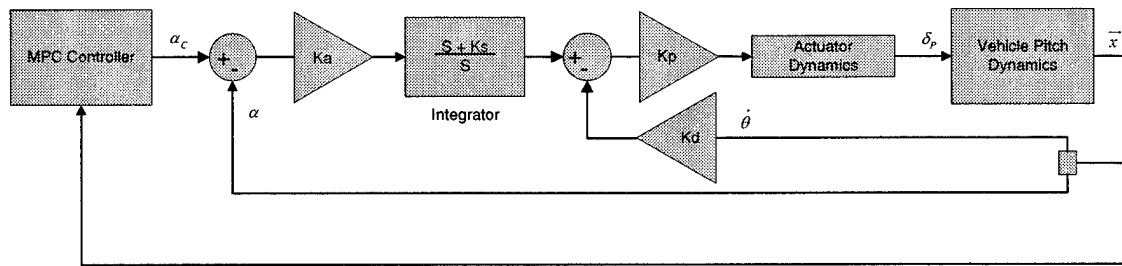
Figure 4-1: Candidate Architecture 1: MPC outer loop with α controlling inner loop

The first architecture is referred to as αSAS and consists of an MPC controller issuing angle of attack (α) commands to an inner loop stability augmentation system

(SAS), which then issues commanded gimbal angle commands (δ_C) to the gimbal actuator. The actuator dynamics then issue the actual gimbal command (δ_p) to the vehicle.
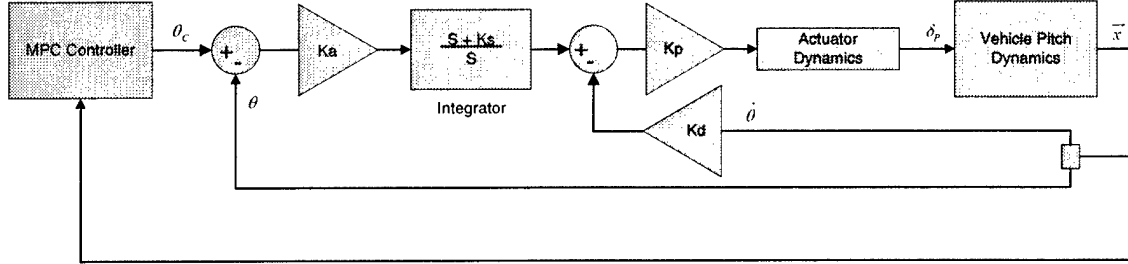
Figure 4-2: Candidate Architecture 2: MPC outer loop with θ controlling inner loop

The second architecture, referred to as θSAS, is very similar to the first, except that the MPC controller issues pitch (θ) commands to an inner loop SAS. This SAS then issues δ_C commands to the actuator dynamics, which in turn issue the true δ_p commands to the vehicle.
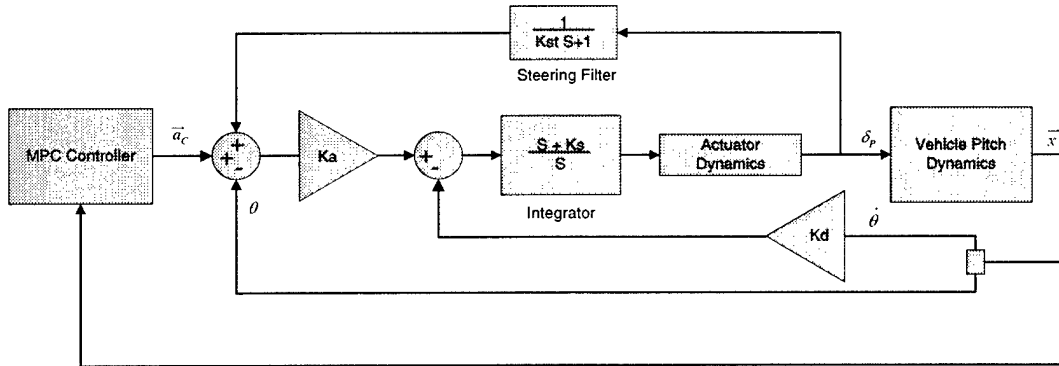
Figure 4-3: Candidate Architecture 3: MPC outer loop with acceleration direction controlling inner loop

The third architecture, referred to as $\vec{a} SAS$, consists of an outer loop MPC controller issuing acceleration direction commands to an inner loop acceleration direction controller SAS. In turn, this SAS then issues δ_C commands to the actuator dynamics, which then issued the final δ_p command to the vehicle.
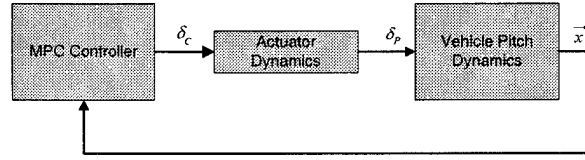
Figure 4-4: Candidate Architecture 4: MPC outer/inner loop

The fourth architecture, referred to as *NoSAS*, was unique among the five controller layouts in that it was the only option in which the MPC controller performed the duties of both the outer guidance loop and the inner stability augmentation loop. The δ_C commands were issued directly to the actuator dynamics, which then issued the actual δ_p commands to the vehicle.
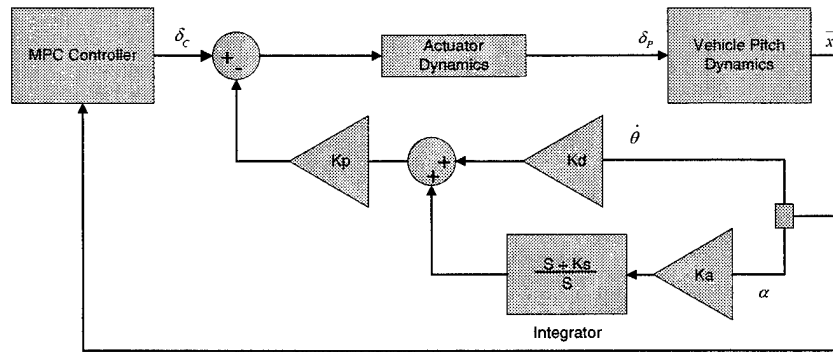
Figure 4-5: Candidate Architecture 5: MPC outer loop with δ controlling inner loop

The fifth architecture, referred to as δSAS, consisted of an MPC controller issuing gimbal angle commands to an inner loop δ SAS. This SAS then issued δ_C commands to the actuator dynamics, which issued the actual δ_p commands to the vehicle.

The performances of all five candidate architectures were compared to the performance of a baseline load-relief controller layout. This controller was designed to represent a current launch vehicle load relief scheme, as detailed by Moreno [5]. A schematic of this baseline controller layout is shown in Figure 4-6.
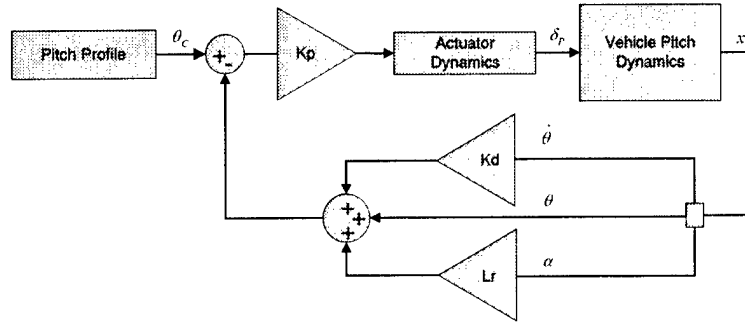
Figure 4-6: Baseline Architecture Layout

The baseline load relief controller was designed to follow a nominal pitch trajectory. If unexpected wind disturbances introduced additional α, the load relief channel would inject a separate signal into the feedback, which would act to bring α back to zero. Because all prospective architectures were rated relative to this baseline, the actual performance of the baseline was less important than the performance of the alternative MPC controllers in relation to the baseline.

The controller gains were chosen to produce controllers with bandwidths of below 2 rad/sec. All gains are unitless except Kd, which has units of seconds. The gains used in these comparisons are shown below.

	αSAS	θSAS	$\vec{a} SAS$	$NoSAS$	δSAS	Baseline
Ka	1	1	0.3	–	1	–
Kd	0.9	0.9	0.35	–	0.9	0.9
Ks	0.1	0.1	0.4	–	0.1	–
Kp	1	1	–	–	1	0.7
Kst	–	–	3	–	–	–
Lr	–	–	–	–	–	0.5

Table 4.1: Architecture Gain Values

Because all vehicle pitch dynamics were represented by linear state-space models, analysis was conducted only around the flight point represented by the linearized model. This point was chosen to be the point of maximum dynamic pressure, which was approximately 80 seconds into a nominal flight. Because the vehicle is most aerodynamically unstable at this flight point, it is normally considered the most

critical design point in a launch vehicle's ascent. Simulations were restricted to a ten second increment of the flight, centered at the design point.

4.2 Linear Vehicle Pitch Dynamics

The linear state-space pitch dynamics model used for this analysis was produced by the same process used to create the plant ID information at every flight point for the non-linear MPC controller. The pitch dynamics represented the vehicle's behavior at approximately 80 seconds into a nominal flight. This LTI model was in the standard state space form shown in Equation 3.1. This model was then reduced to only contain the α, $\dot{\theta}$ and θ states. In addition, the effects of a wind disturbance on the states and output were included in the linear model via the appropriate B_v and D_v measured disturbance matrices.

4.3 Wind Models

Both synthetic and actual wind samples were used for this comparison. Actual wind profiles were sampled at the appropriate altitude for the time period during launch associated with the vicinity of the flight point to produce the actual wind samples. Synthetic wind samples were produced by passing a random signal through a low-pass filter which produced a simulated wind disturbance profile. This wind and an appropriate vehicle speed were then used to calculate the α that a vehicle flying through this wind would experience. Every random profile was linked to a unique random number or "seed", which was used to initialize the random signal generator.

4.4 Architecture Comparison Results

The results shown in Figure 4-7 represent the analysis of 1000 synthetic wind profiles and 127 actual wind profiles. Options 1 - 5 represent the αSAS, θSAS, $\vec{\alpha} SAS$, $NoSAS$ and δSAS architecture options, respectively.
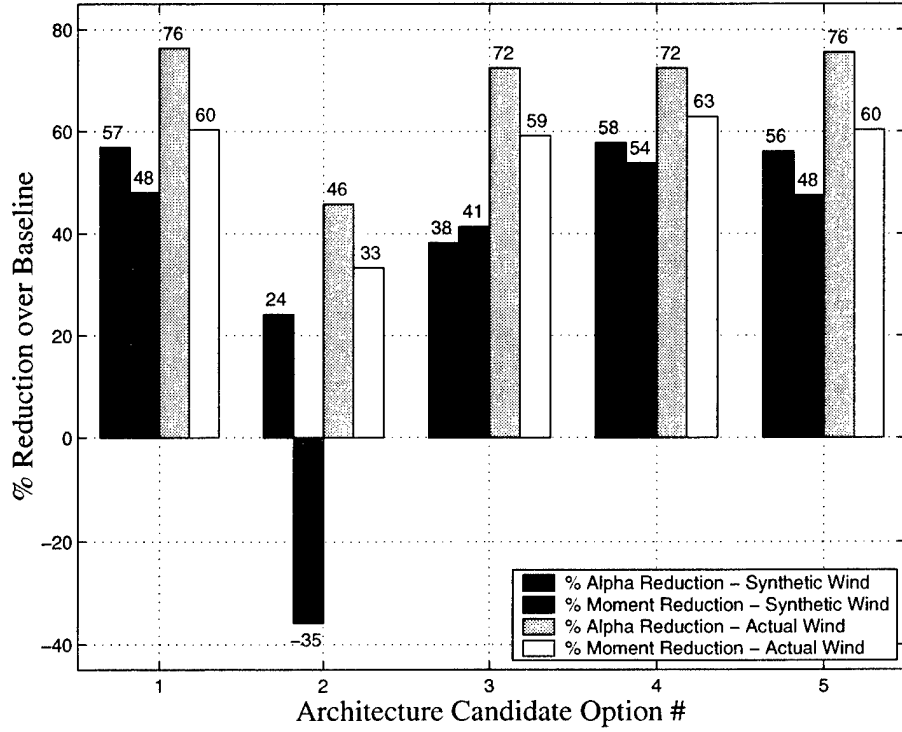
Figure 4-7: Architecture Comparison Results

The results from these trials indicate that the θSAS option was clearly deficient in both α reduction and bending moment reduction as compared to the other four options. The $\vec{a} SAS$ showed acceptable performance in α reduction, but performed slightly worse than the remaining three options in bending moment reduction. The $NoSAS$ option showed the best overall reduction of bending moments, as well as excellent α reduction capability. Because of this performance and its unique control layout, this architecture was selected for further investigation. In addition to the non-SAS approach, a inner-loop SAS approach was desired to serve as an additional comparison case. While the remaining two architectures, δSAS and αSAS, showed nearly identical results, the αSAS option was chosen because of its more familiar and intuitive controller layout. Unlike these two controller architectures, which both featured α feedback, the θSAS and $\vec{a} SAS$ options seem to have been hampered by the lack of direct α feedback. The direct control of α was considered advantageous in terms of both controller integration and trajectory design integration.

66

The results of this down-select process were a SAS equipped and a non-SAS equipped MPC control layout. These two architectures would allow the research to investigate the advantages and disadvantages of each type of system in relation to the current load-relief standard, while providing further insight into the design process that applies to each approach. For the remainder of this thesis, the MPC controller with an inner-loop α controller will be referred to as *ASAS*, and the MPC controller with no inner-loop SAS will be referred to as *NSAS*.

[This page intentionally left blank.]

Chapter 5

Wind Discussion

5.1 Wind Profiles

The wind profiles used for this research were gathered from the Eastern Test Range (ETR), located at Kennedy Space Center, FL, and the Western Test Range (WTR), located at Vandenburg AFB, CA. All wind profiles were gathered using jimsphere wind observation methods. Reflective aluminum balloons, such as the one shown below, are released and tracked by radar as they ascend through the atmosphere. The radar track is then analyzed to calculate the wind speed at each altitude.
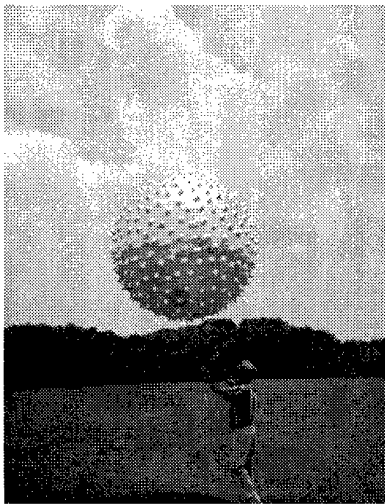
Figure 5-1: Jimsphere Wind Measurement Balloon

The ETR wind set included 1,927 wind profiles, recorded between the years 1964 and 2000. The WTR wind set included 1,437 wind profiles, recorded between 1965 and 1980. Although some of the profiles were unusable, due to tracking and transmission errors, a large majority of the wind profiles, 1,141, were judged to be accurate and realistic representations of the actual wind state at the time of the observation. The wind data gathered from both ranges included wind speed and wind direction, in roughly 80 foot increments, from the surface to an altitude of approximately 95,000 feet. Because all research was conducted in the pitch plane only, it was necessary to decompose the wind velocity into North/South and East/West components. Because the WTR is primarily used to launch payloads into polar orbits, all tests using WTR wind profiles were conducted using the North/South wind component. In the same way, all ETR tests were completed using the East/West wind component, as the ETR is primarily used to launch payloads into equatorial orbits. Figure 5-2 shows a single representative wind profile from the ETR.
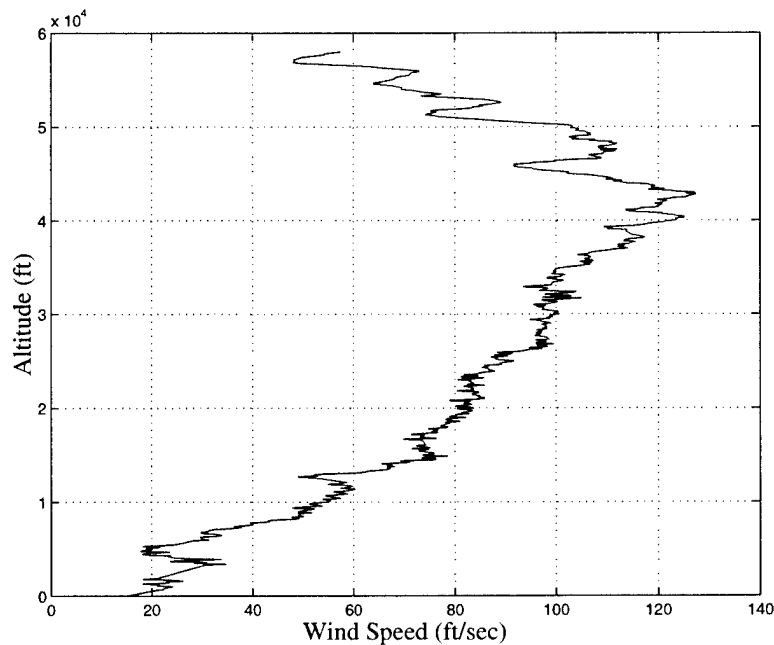
Figure 5-2: Sample Wind Profile

Figure 5-3 shows all 127 ETR wind profiles used in the MPC controller design

process. This set will hereafter be called the design wind set. This wind set is a representative sample of the complete wind set used in the actual system comparisons.

The large number of wind profiles capture a broad cross-section of weather types and wind behavior, including headwind and tailwind conditions. This wide range of wind conditions presents a load relief problem for the control system. Load relief is most critical during portions of ascent where dynamic pressure (Q) is high. During this time, all MPC load relief controllers seek to minimize vehicle α and thus $Q\alpha$, the product of Q and α. Because α is directly affected by wind speed and direction, only one unique pitch trajectory exists which will produce the minimum α for any given wind profile while avoiding excessive Q levels. To find this unique pitch trajectory for each wind profile, a Matlab Trajectory Designer was created.
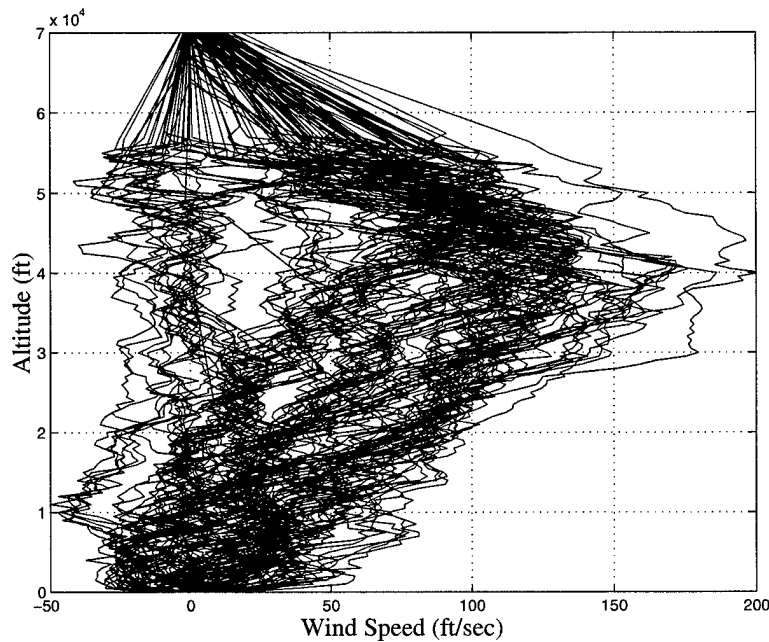
Figure 5-3: ETR Wind Profile Set

5.2   Trajectory Design

Because of low vehicle speed in the early seconds of flight, the vehicle must be guided by a pitch controller until it has reached a safe altitude and speeds great enough

to require α control. In addition, because of the low vehicle speeds, no significant load relief benefits are gained by minimizing α at low speeds. Because of this, all launch vehicle trajectories in this research can be divided into two major portions: the pitch control portion and the α control portion. To quantify each trajectory further, the pitch control portion can be divided into three more sections. The first section consists of a vertical rise from the launch pad, for some designated rise time, T_r. After this initial rise, the vehicle enters the second section, in which it executes a kick maneuver to some designated pitch angle, $kick\theta$, over some designated kick time, T_k. Following this, the vehicle enters the third portion of the flight, where it flies at the constant pitch angle, $kick\theta$, until a given transition Q level is reached. At this point, the vehicle transitions into load-relief control and enters the fourth portion of the trajectory, which lasts for the remainder of the boost phase. Load relief is achieved in the initial trajectory design process using a standard PID α controller, in the form illustrated in Figure 6-1.

Each acceptable trajectory must meet several end conditions. The K-1 has a nominal boost phase flight time of 134 seconds. At the end of the initial boost phase, vehicle staging requirements dictate that the vehicle must be at a staging Q of approximately 60 lbf/ft^2 (psf). A third, implicit requirement states that the vehicle must not experience a Q of more than 600 psf at any point during the initial boost phase. The Trajectory Designer works to meet the staging dynamic pressure constraint by iteratively adjusting the parameters T_k and $kick\theta$ until a 134 second trajectory is found which produces a final Q that falls within 2 psf of the staging requirement. If both of the explicit requirements are met, the implicit maximum Q limit will not be exceeded. The trajectory design process is illustrated in Figure 5-4.

Because the wind profiles only extend to altitudes of approximately 55,000 feet, each profile was augmented with extra wind data to produce a smooth transition from the final wind speed to zero, as seen in Figure 5-3. However, because all load relief comparisons are completed below the augmented data, this contrived wind data does not present a problem with simulation accuracy. The additional wind information is used only by the Trajectory Designer and has no effect on the pitch trajectory during

Staging Conditions
Q = 60 psf
t = 134 sec

Gravity Turn
$\alpha = 0$
Load Relief
SEGMENT 4

Trajectory defined in terms of four segments:
- SEGMENT 1: Vertical Rise
- SEGMENT 2: Kick Maneuver
- SEGMENT 3: Constant Attitude
- SEGMENT 4: $\alpha = 0$: Load Relief

Trajectory Designer seeks to meet Staging Conditions by iterating:
- kAngle (kickθ)
- Kick Duration (T_K)

Constant Attitude
SEGMENT 3

$Q > Q_{Transition}$

kAngle

Kick End = $T_R + T_K$

Kick Maneuver
SEGMENT 2

Kick Duration = T_K

Launch Site

Kick Start = T_R

Vertical Rise
SEGMENT 1

Figure 5-4: Trajectory Design Framework

the load relief portion of the flight that is measured.

Once an acceptable trajectory was found, the pitch profile was saved for later use, as well as the appropriate trajectory parameters and other important profiles, such as Q. Each data set was indexed to the particular wind profile to which it applied. It is also important to note that, while the trajectories produced by the Trajectory Designer are very close approximations to actual KAC pitch trajectories, they are not accurate enough to be used for actual launch operations. However, they are qualitatively identical to real launch trajectories and serve as valid approximations for this research, which is not guidance oriented, but rather load relief oriented.

Figure 5-5 shows the θ trajectory from a sample wind profile. Notice the distinct portions of the flight, marked by distinct changes in pitch.
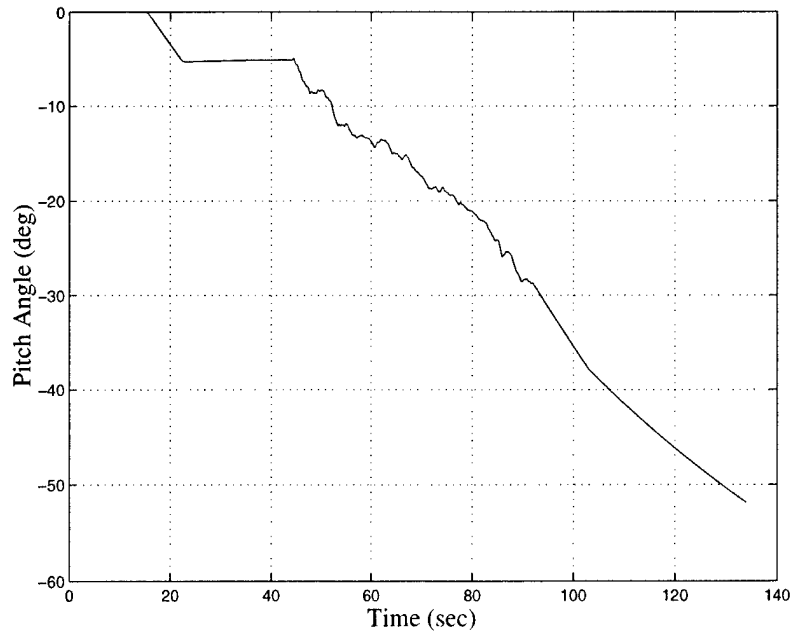
Figure 5-5: Sample Pitch Trajectory
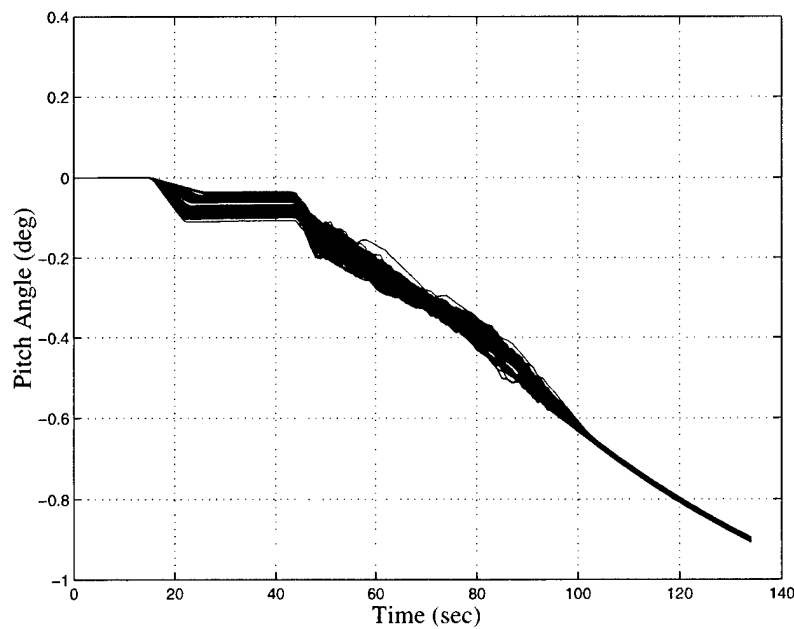
Figure 5-6: ETR Pitch Trajectory Set

Figure 5-6 shows a plot of all the pitch trajectories designed for the design wind set. Notice the wide range of T_k and $kick\theta$ variables present in response to the varying

wind conditions contained within the wind set.

Figure 5-7 shows a plot of all the Q profiles resulting from the pitch trajectories shown in Figure 5-6. Notice the convergence of the profiles to 60 psf at the staging time of 134 seconds.
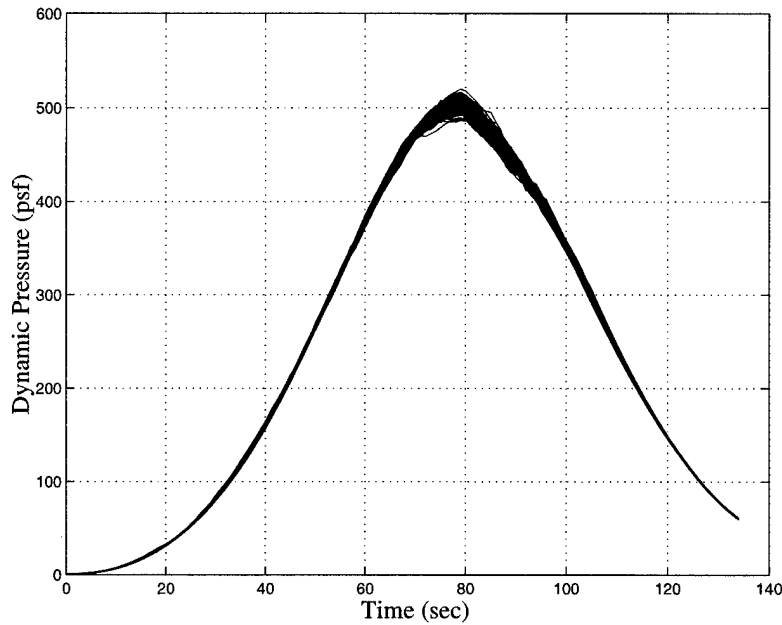
Figure 5-7: ETR Q Profile Set

Equivalent trajectories, although not shown, were designed for the remaining 1,800 ETR wind profiles and 1,437 WTR wind profiles.

5.3 Time Lapse Discussion

The current method of pre-launch wind measurement involves the release and tracking of multiple jimsphere balloons, as discussed previously. Delays caused by slow balloon ascension and data processing are significant. The standard wind profiles that are loaded into a current launch vehicle load-relief system were usually measured at least 90 minutes before the vehicle is scheduled to launch. This delay introduces large amounts of uncertainty into the launch decision process, and can often result in cancellations or delays due to quickly changing wind conditions. To simulate this

delay, the time stamp of each wind profile was analyzed to find the wind sets with time differences of 90 minutes, ± 15 minutes. The actual wind profiles from these sets were used to simulate a traditional balloon-based wind measurement system integrated with an MPC controller or a modern traditional load-relief system. Below is a sample wind pair selected from the ETR.
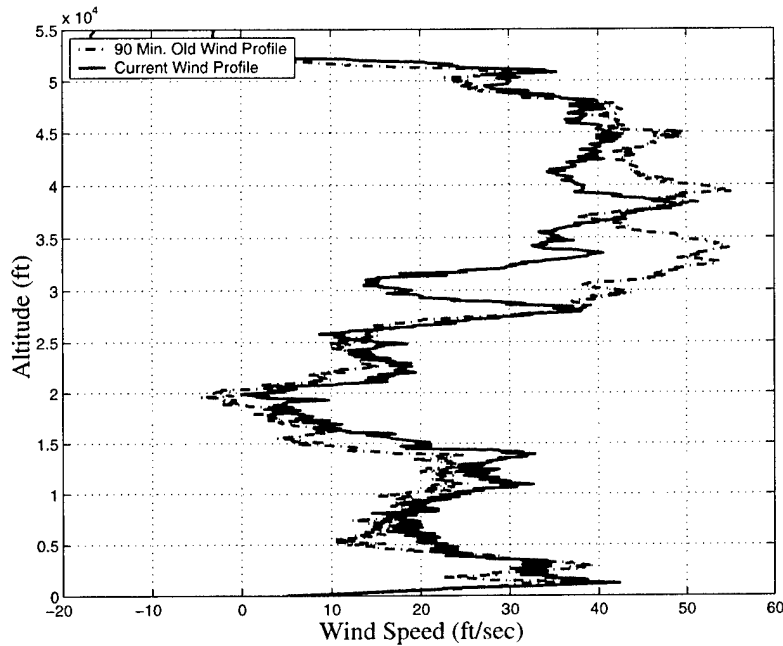
Figure 5-8: Sample ETR Wind Pair

Some current load-relief systems do not use the actual wind profile during flight, but rather a polynomial fit of each wind profile. Depending on the system, they assume the wind follows a 6th or 15th order polynomial fit, and then design a trajectory to minimize bending moments or α for that modified wind profile. Representing these systems required a special trajectory design process. Depending on the system fidelity, the Trajectory Designer would replace the actual wind profile with a polynomial fit of the appropriate order, and then complete the trajectory design process as described previously. Thus each wind pair would contain three different pitch trajectories, each corresponding to a different fidelity of wind profile representation.

During time lapse system comparison, the more recent profile in each wind pair

would be loaded into the simulation as the actual wind, while the appropriate pitch trajectory designed for the earlier wind profile, depending on the control system being simulated, would be loaded into the vehicle control system. Below is a comparison of a sample wind profile and its 6th and 15th order polynomial fits. The complete results of these comparisons are included in Chapter 8.
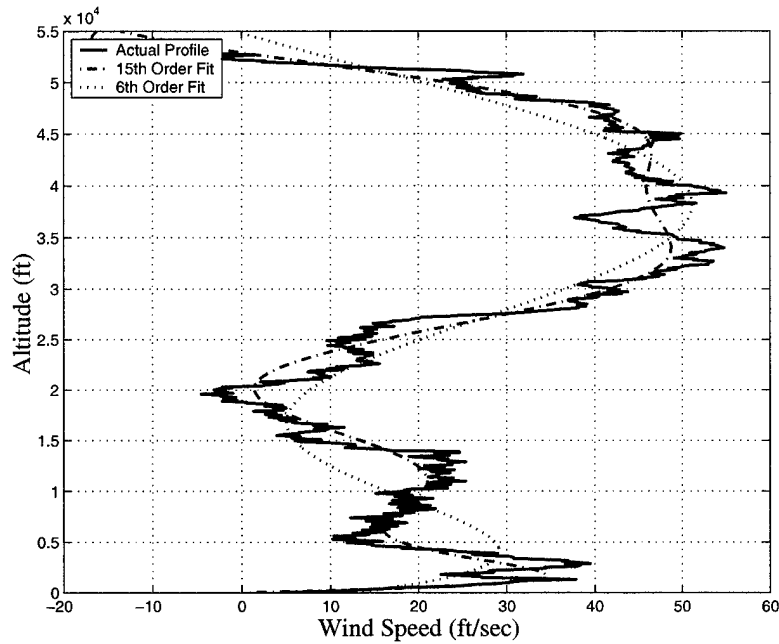
Figure 5-9: Comparison of Actual Wind Profile, 6th Order Fit and 15th Order Fit

[This page intentionally left blank.]

Chapter 6

Baseline Controller Development

6.1 PID Controller Description

All PID controllers used in the vehicle simulations followed a standard layout. This type of controller was chosen for several reasons. Primarily, this controller type was chosen because it was somewhat similar to the controller used in the actual K-1 launch vehicle. The position of the four gains also allows the controller to handle derivative rate feedback and address steady state error, while also allowing a high degree of flexibility in the design process. This layout was used in all simulations to control vehicle yaw attitude. In addition, this controller layout was used to control vehicle pitch attitude in the baseline vehicle simulation. Finally, this layout served as the inner-loop, α controller layout for the SAS-equipped MPC vehicle simulation.
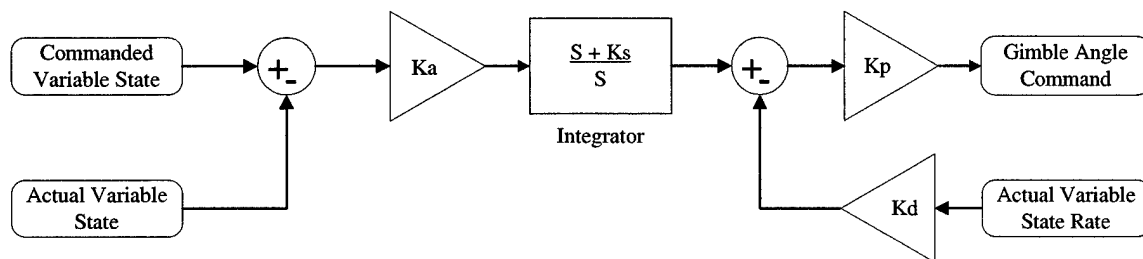
Figure 6-1: Standard PID Controller Layout

Four gains, Ka, Kd, Kp and Ks, were employed with a lead compensator and an integrator as shown in Figure 6-1.

6.2 Controller Gain Selection Method

The gain selection process was considered essential in building a fair comparison structure between the baseline simulation and the MPC-equipped simulations. The main method used to ensure fair controller comparisons between drastically different controller architectures was to match the closed loop bandwidths of the controller/vehicle systems. A system bandwidth of 1.75 rad/sec was chosen as a target for the controller gain selection process. Restricting the bandwidth allows higher order effects such as fuel slosh, bending modes, Tail-Wags-Dog effects, computational delays and transport delays to be neglected without compromising the validity of the simulation. Because the vehicle dynamics change significantly throughout a standard flight, two of the four controller gains must be dynamically scheduled. A Matlab script file was created to calculate the gains which would yield the appropriate controller bandwidth at each flight point. Flight points were defined at one second increments throughout the flight.

In the interest of process speed, a Matlab search algorithm, *fminsearch.m*, was used to conduct the gain selection. While holding Ka and Ks steady at 1.4 and 0.1 respectively, the algorithm adjusted the values of Kd and Kp within a dedicated SIMULINK model containing the controller and plant pitch dynamics. During each search iteration, this file performed a linearization of the model and then calculated the phase and magnitude response of the combined linear model to a range of input frequencies. A linear interpolation was then used to find the closed loop bandwidth, defined as the frequency at which the closed loop magnitude response fell below -3 dB. This process was repeated until the error between the system bandwidth and the target bandwidth fell below an acceptable tolerance level, in this case 0.01 rad/sec. A plot of the resulting gain schedules is shown in Figure 6-2, followed by a plot of the resulting closed loop system bandwidth at each flight point, in Figure 6-3.
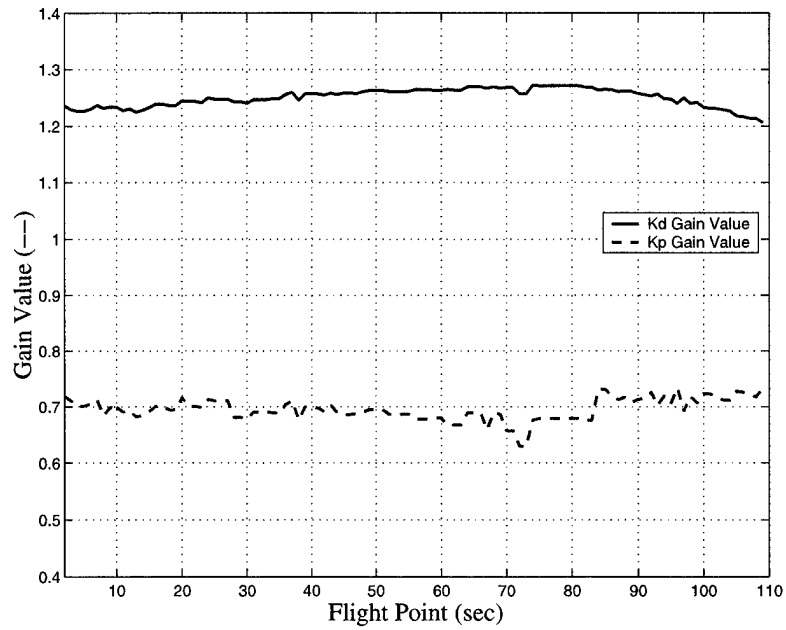
Figure 6-2: Scheduled Gain Values at each Flight Point
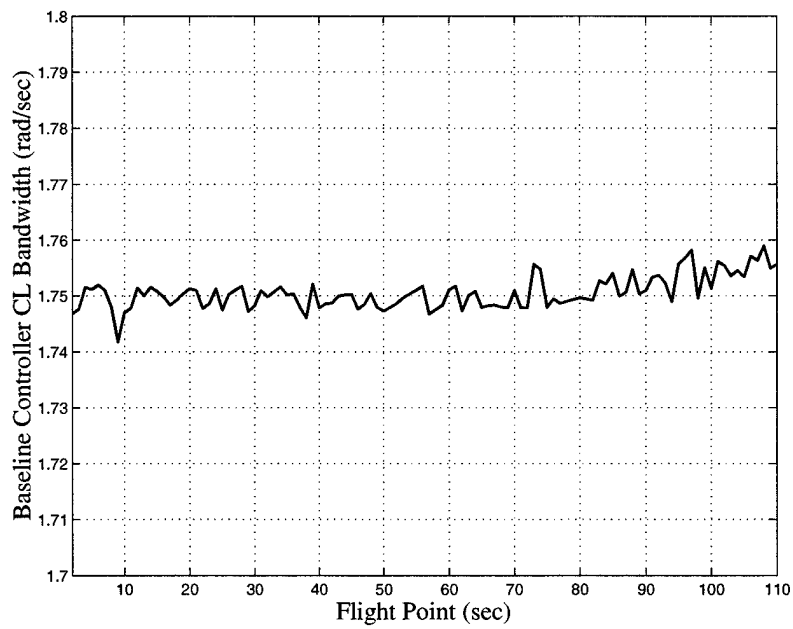
Figure 6-3: Closed Loop Bandwidth of Baseline Controller at each Flight Point

81
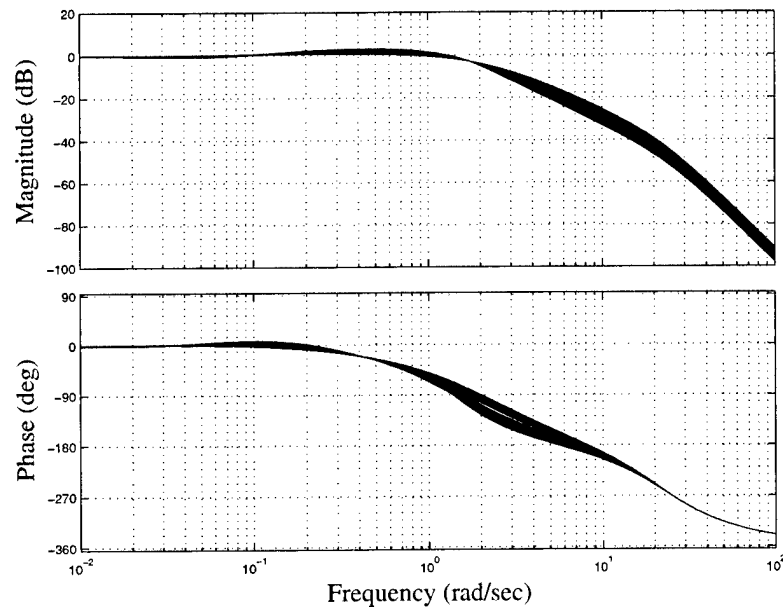
Figure 6-4: Closed Loop Bode Plot of Baseline Controller at each Flight Point

A bode plot of the system at each flight point is shown in Figure 6-4. Notice the 1.75 rad/sec frequency, where all magnitude responses converge at the -3 dB line. Although not explicitly included in the search algorithm, gain and phase margin requirements of 6 dB and 30 deg, respectively, were implicitly imposed on the resulting systems. The gain and phase margin of the closed-loop system is shown in Figures 6-5 and 6-6 respectively. With an average gain margin of more than 22 dB and an average phase margin of more than 100 deg, the controller far exceeded the stability requirements at all flight points.

While no explicit system performance requirements were imposed upon the gain selection process, the performance of the controller at each flight point was evaluated to ensure that large performance gaps did not exist between the baseline controller's performance and standard "acceptable" performance criteria. Figure 6-7 shows a plot of the step response of the controller at each flight point. The average percent overshoot was less than 20%, while the average rise time was about 1.3 seconds, both well within acceptable limits.
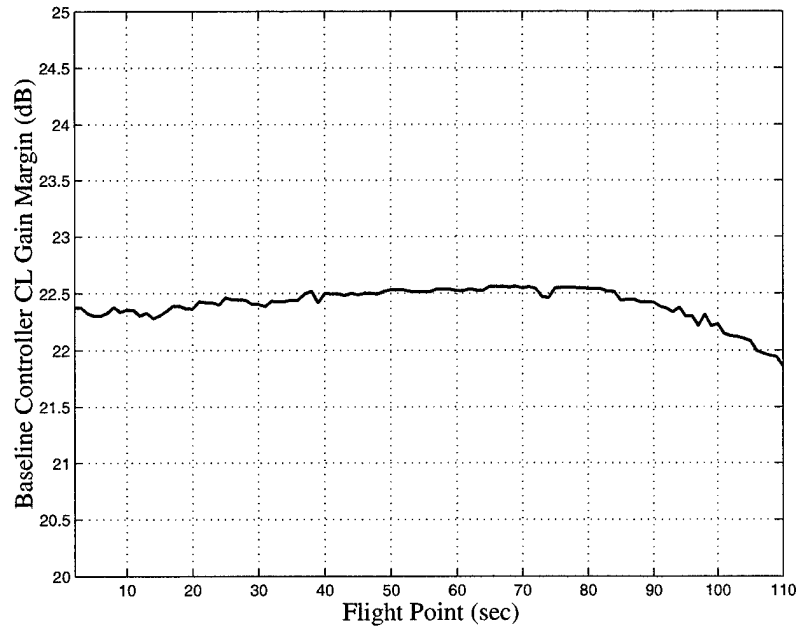
Figure 6-5: Gain Margin of Baseline Controller at each Flight Point
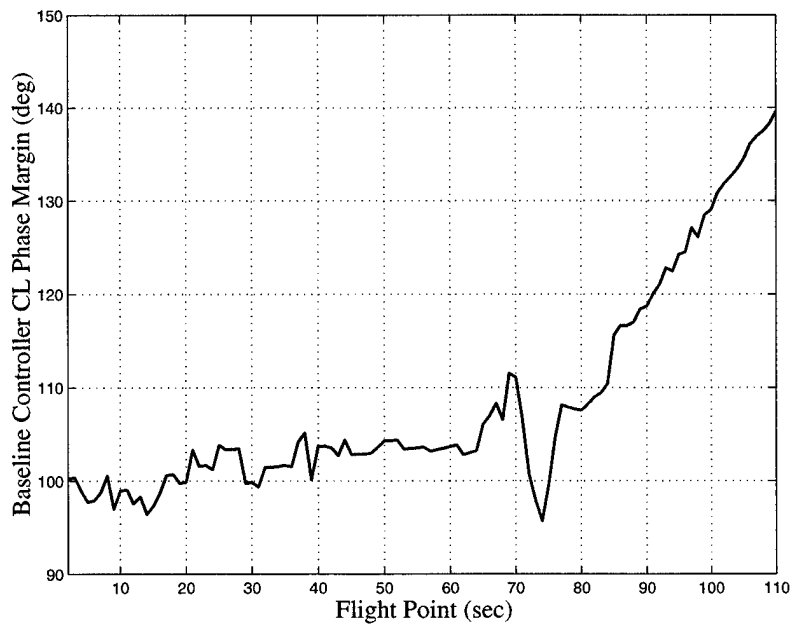
Figure 6-6: Phase Margin of Baseline Controller at each Flight Point

This method of determining controller gains produced a control system with a narrowly constrained bandwidth throughout the entire flight envelope. This allowed
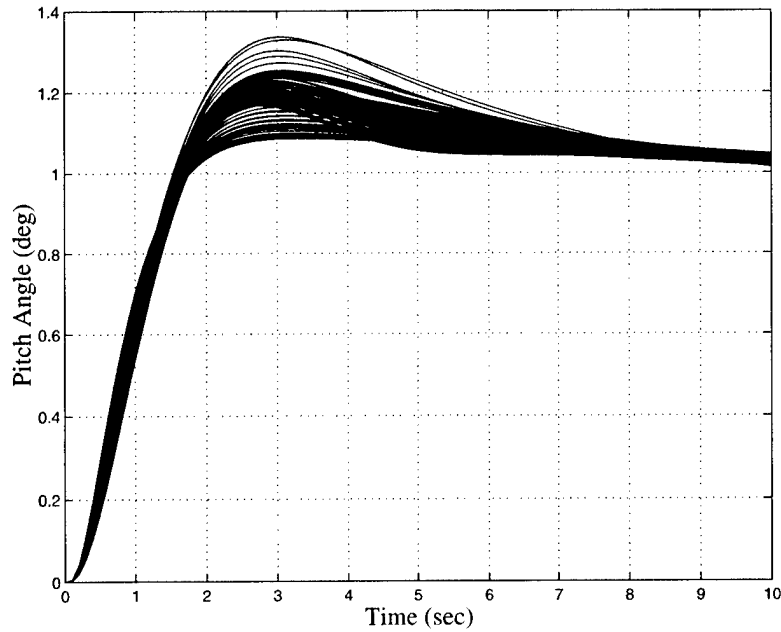
Figure 6-7: Step Response of Baseline Controller at each Flight Point

fair performance comparisons between this controller and other controllers that were similarly constrained, while still maintaining adequate performance and minimum required levels of stability.

Because the yaw controller was not required to handle any disturbances or non-zero trajectory commands, and no analysis was conducted in the yaw plane, fixed controller gains were considered sufficient for all simulation yaw controller gains. These gains were set equal to the scheduled pitch controller gain values at the flight point nearest maximum Q.

Chapter 7

MPC Controller Development

7.1 Controller Design

The design process for a MPC controller requires defining multiple design parameters. These parameters include the controller cycle rate, the prediction step size, the W_y, W_u, and W_Δ weighting matrices, the J_M matrix and the prediction horizon. Presently, there is no set procedure which can be followed to design an MPC controller, as there is for an LQR controller, for example. This chapter will attempt to define such a process, by describing the steps which were used to set each of the parameters in the research controllers. While still requiring engineering intuition and experience, this process will serve to build a basic framework or road map, which can be followed when designing any MPC controller. In addition, the actual implementation of the MPC controller in SIMULINK will be described. All simulations included in this chapter were conducted using the design wind set, as detailed in Chapter 5.

7.1.1 MPC Controller Range

It is important to note that the MPC controller was only designed to operate between 40 and 100 seconds in a nominal launch profile. As described in the Trajectory Design section of Chapter 2, the first portion of the boost profile is executed using a standard pitch controller, following a pre-designed pitch profile. At a certain Q level,
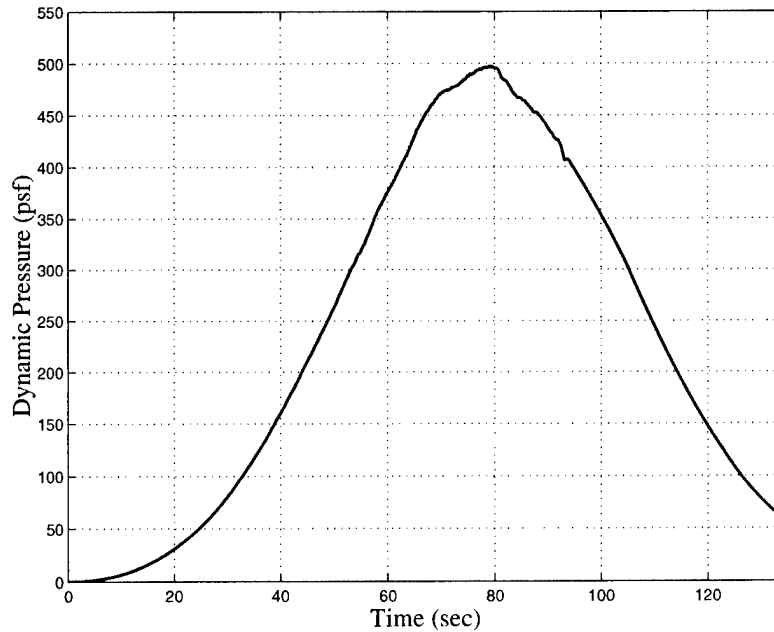
Figure 7-1: Standard Dynamic Pressure Profile

control transitions from the standard PID pitch controller, described in Chapter 6, to the MPC controller, which seeks to minimize α. Because of the Q levels chosen as transition points, this control hand-off always occurs after 40 seconds into the flight. To save time, no MPC control information was calculated before this point. In addition, all wind profiles used as disturbance models terminate at altitudes of approximately 55,000 ft, which occurs at about 95 seconds into a nominal flight. At this altitude, Q has fallen below the critical level, due to a rapid decrease in atmospheric pressure. Because of this, terminating the trajectory comparisons at 95 seconds into each flight was not considered problematic. Thus, no MPC control information was calculated after 100 seconds. Figure 7-1 shows a plot of a Q curve from a standard boost trajectory. All system comparisons were completed within the high Q region from approximately 50 seconds to 95 seconds.

7.1.2 MPC Design Process Overview

The steps used to design the *ASAS* and *NSAS* MPC controllers are shown in the following list:

1. Choose desired system bandwidth

2. Choose magnitude of look-ahead time

3. Choose relative gains of W_y, W_u, and W_Δ weighting matrices

4. Choose prediction step size

5. Choose J_M matrix form

6. Choose controller rate

This design process will not exhaustively cover the entire design space. This approach is not feasible, due to the extremely high number of design parameters, and the wide range of possible values for each parameter. Instead, this process will focus on a sequence of steps which reduce the range of possible controller designs in an orderly fashion, while providing satisfaction that pertinent design possibilities have not been overlooked.

7.1.3 System Bandwidth

As detailed in Chapter 6, a constant system bandwidth of 1.75 rad/sec was chosen to ensure a fair comparison between the MPC controllers and the baseline PID controller. This target bandwidth is representative of the control system bandwidths found in multiple contemporary launch vehicles and allows higher-order dynamics to be neglected without compromising the validity of the simulation. The bandwidth choice affects all following performance metrics and controller parameters, and may require iteration to address the unique characteristics of certain launch vehicles. A discussion of the effects of the closed-loop bandwidth on this vehicle is included at the end of this section.

7.1.4 Look-Ahead Time Magnitude

The next step of the design process is to choose the look ahead time that the MPC controller will use in its computations. For this step, the input weighting matrices, W_u, and W_Δ, were set to identity matrices, leaving only the output matrix, W_y, to be manipulated. This allowed the simple manipulation of the ratio between the input and output weighting matrices to affect the system closed loop bandwidth.

A Matlab search algorithm, *fminsearch.m*, was used to conduct the W_y gain selection. While holding the W_u and W_Δ gains constant at unity, the algorithm adjusted the value of the W_y gain within a dedicated SIMULINK model containing the controller and plant pitch dynamics. During each search iteration, this file performed a linearization of the model and then calculated the phase and magnitude response of the combined linear model to a range of input frequencies. A linear interpolation was then used to find the bandwidth, defined as the frequency at which the magnitude response fell below -3 dB. This process was repeated at each appropriate flight point until the error between the system bandwidth and the target bandwidth fell below an acceptable tolerance level, in this case 0.01 rad/sec. This process was completed for both MPC control architectures over a range of prediction step sizes and look ahead times. Figures 7-2 to 7-7 show plots of the W_y gain schedules for three selected prediction step sizes for each MPC architecture. Look ahead times of 1 to 10 seconds were analyzed. Figure 7-8 shows all the resulting system bandwidths. All bandwidths fell within the prescribed tolerances.

These graphs show that the gain schedules are very different for look ahead periods of 1 to 3 seconds than they are for look ahead periods of 4 seconds and greater. Above 4 seconds, the gain schedules show very little change as the look ahead period is increased. This convergence behavior implies that increasing the look ahead period beyond 4 seconds will provide very little change in performance. However, the computational load will continue to rise in proportion to the square of the look ahead period, due to the structure of the MPC matrices. To show that increasing the look ahead period beyond 4 seconds does not yield a significant benefit in system
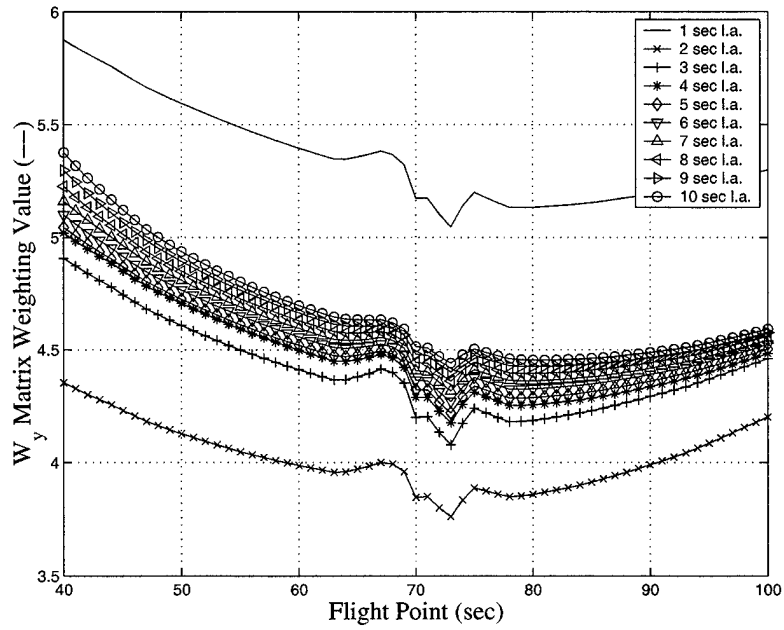
Figure 7-2: W_y Gain Schedules for $ASAS$, Prediction Step $= 0.2$ sec
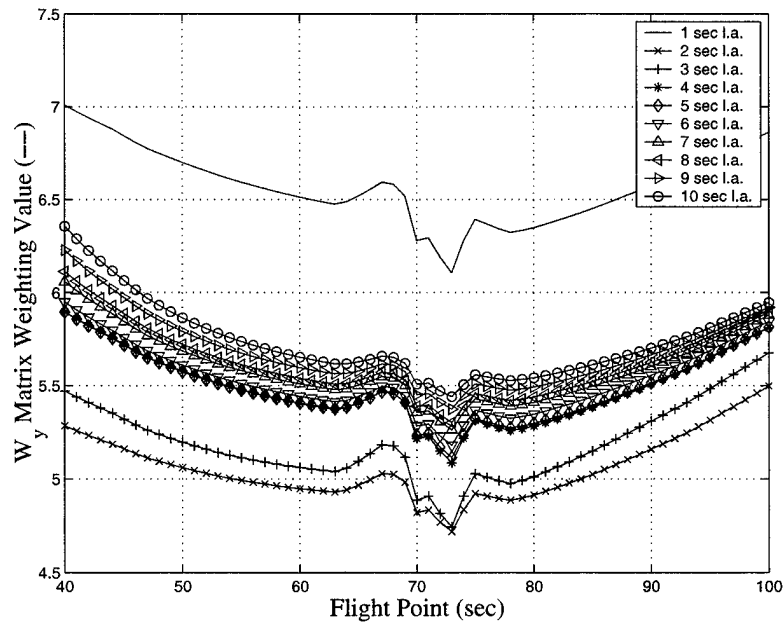
Figure 7-3: W_y Gain Schedules for $ASAS$, Prediction Step $= 0.6$ sec
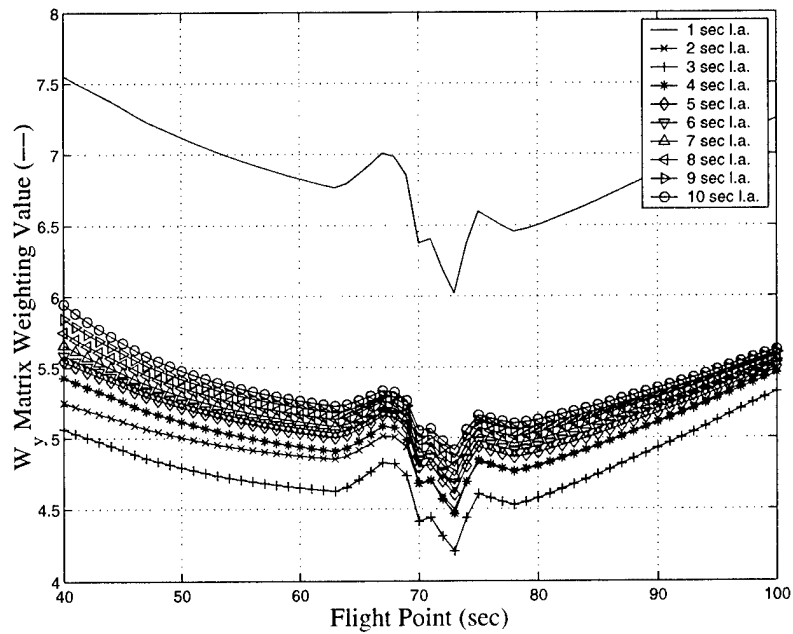
Figure 7-4: W_y Gain Schedules for $ASAS$, Prediction Step $= 1.0$ sec
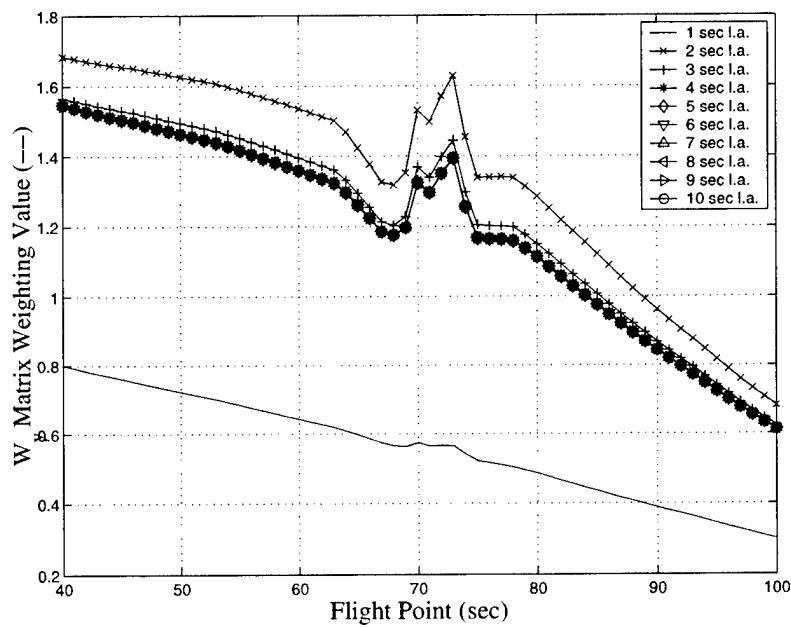
Figure 7-5: W_y Gain Schedules for $NSAS$, Prediction Step $= 0.1$ sec
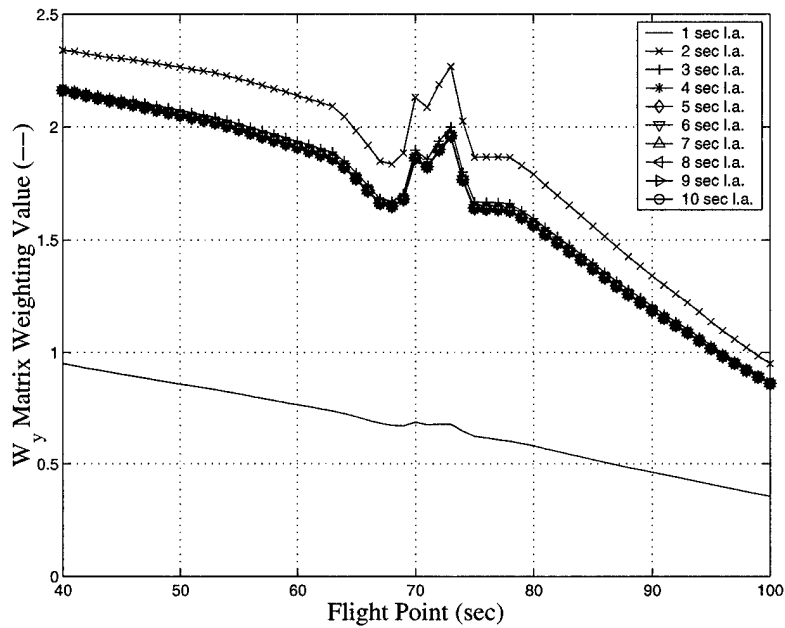
Figure 7-6: W_y Gain Schedules for $NSAS$, Prediction Step $= 0.2$ sec
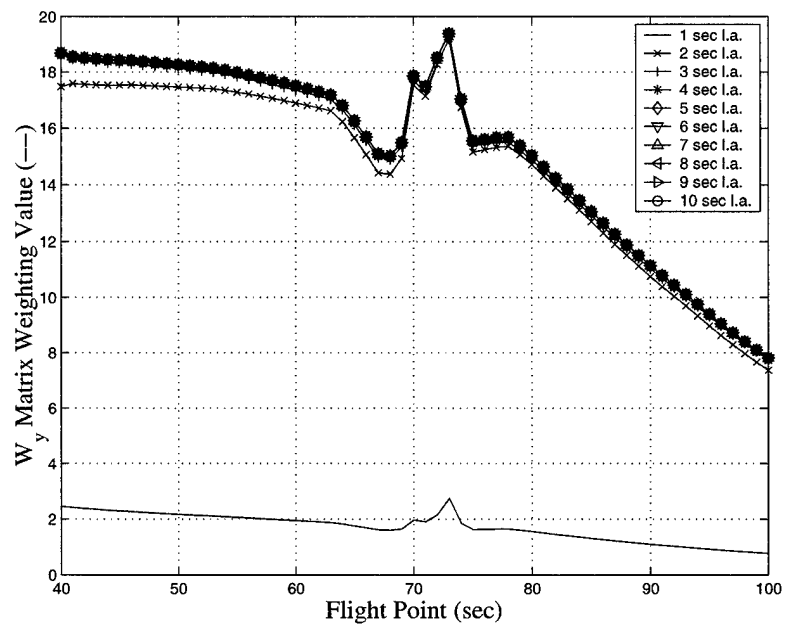
Figure 7-7: W_y Gain Schedules for $NSAS$, Prediction Step $= 0.5$ sec
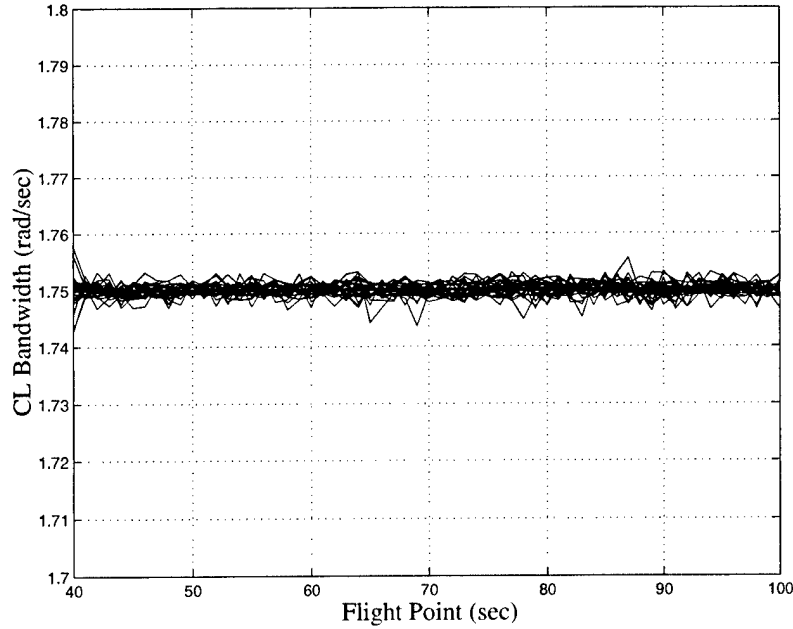
Figure 7-8: Closed Loop Bandwidths for all Controller/Look Ahead Combinations

performance, a set of comparison runs was completed using the design wind set. The controller in the first run used a look ahead period of 4 seconds, while the second used a look ahead period of 10 seconds. For this trial, the prediction step size and controller rate were both set to 0.2 seconds, and the J_M, W_u and W_Δ matrices were set to identities. The results for this trail are shown in Table 7.1.

MPC Controller Type	No SAS	No SAS	α SAS	α SAS
Look Ahead Period (sec)	4	10	4	10
Average Max α (deg)	0.36046	0.36053	0.35713	0.35585
Average Max Qα (psf-deg)	153.28	153.32	151.31	150.64
Average Max Bending Moment (ft-lb)	155687	155628	216323	216139
Average Max Gimbal Angle (deg)	0.37005	0.37026	0.56549	0.57146

Table 7.1: Performance Comparison of MPC controllers with varying look ahead periods

More than doubling the prediction horizon from 4 seconds to 10 seconds has a negligible effect on the metrics shown above, usually resulting in a less than one percent increase or decrease in a given variable. The fact that there were no significant

differences between the two trials shows that increasing the look ahead period beyond 4 seconds does not yield any appreciable increase in performance. Because of this, the look ahead period for both controllers was set to 4 seconds.

7.1.5 W_y, W_u, and W_Δ Matrix Weighting

When choosing the weighting matrices W_y, W_u, and W_Δ, it is important to recognize that the significant design factors are the ratios between the three matrices, and not the actual values of each matrix. This is because MPC seeks to minimize the magnitude of a cost function. Thus, one matrix may be set to unity, leaving only two matrix gains to be manipulated in the design process, without sacrificing any design fidelity. The W_y gain will continue to be manipulated as described previously to match the system closed-loop bandwidth with a given target bandwidth. To examine the impacts of possible W_u and W_Δ gain ratios, a subset of all the possible gain ratios deemed to span a significant portion of the design space was examined. For each architecture, the W_u gain was set to unity, and four different gains were applied to the W_Δ matrix: 0, 1/2, 1 and 2. Simulations using the design wind set were conducted with controllers using these four gain ratios to discover the sensitivity of controllers with each gain ratio to noise in the wind disturbance measurement process. Figures 7-9 to 7-13 contain graphs showing the sensitivities of each architecture/gain ratio combination to random wind disturbance measurement inaccuracies with increasing standard deviations, ranging from 0 to 20 ft/sec. A wind disturbance measurement inaccuracy was defined as the difference between the wind profile used to calculate the MPC control and the wind profile that actually affected the vehicle during flight. For all test runs, prediction step size and controller rate were set to 0.2 seconds and look-ahead time was set to the previously chosen value of 4 seconds. In addition, the effect of changing the weighting on errors as a function of their location within the prediction horizon was not investigated. All errors, regardless of their distance from the present time step, were weighted equally. Thus, each weighting matrix contained only one non-zero value, which was repeated at all points on the matrix diagonal.

Several interesting trends exist in the data. Both architectures exhibit similar
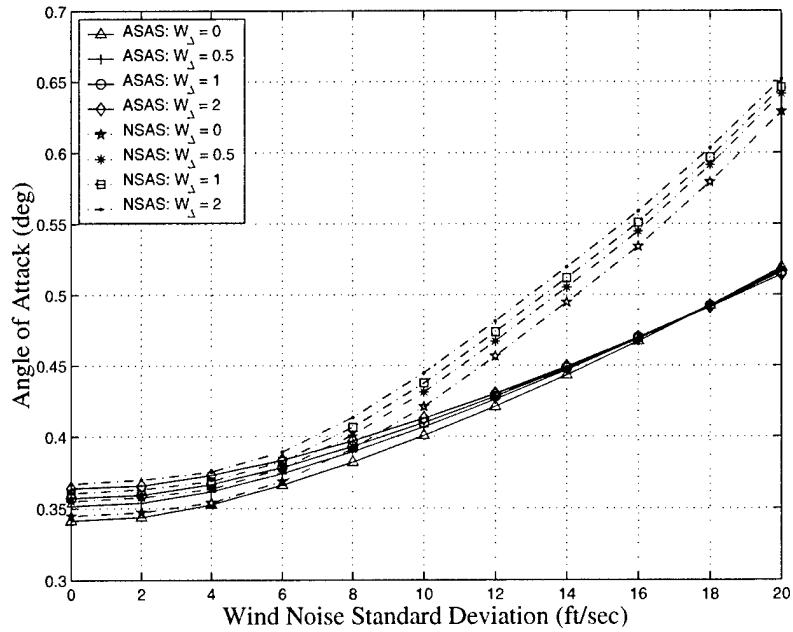
Figure 7-9: Mean Peak α Sensitivity to W_Δ Weighting
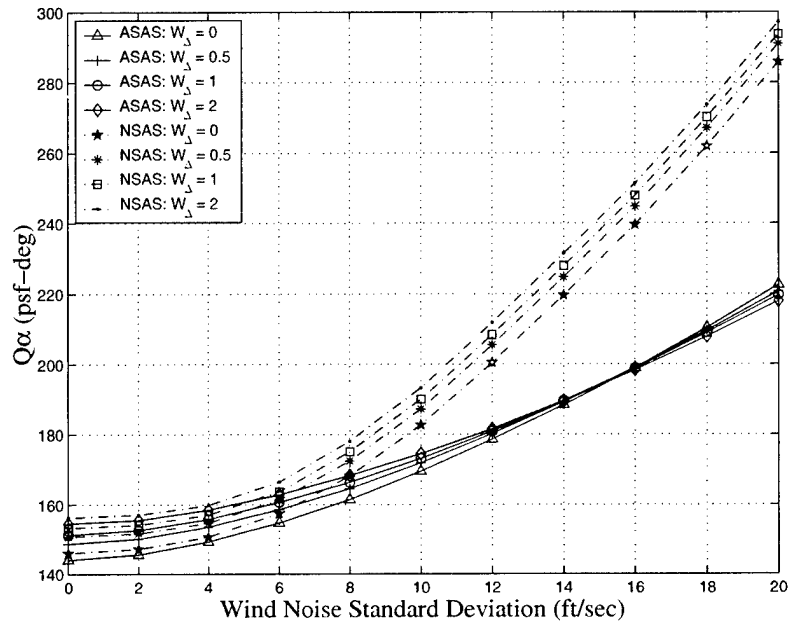
Figure 7-10: Mean Peak $Q\alpha$ Sensitivity to W_Δ Weighting
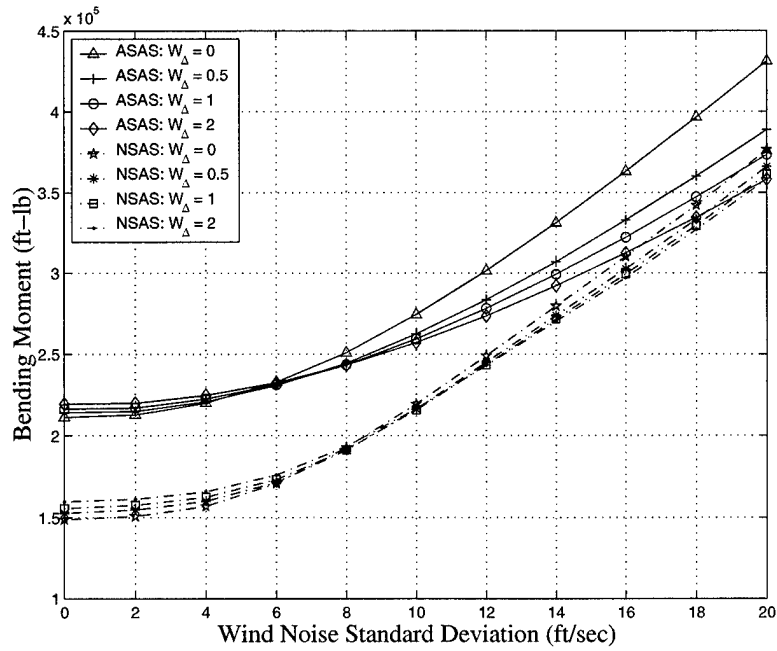
Figure 7-11: Mean Peak Bending Moment Sensitivity to W_Δ Weighting
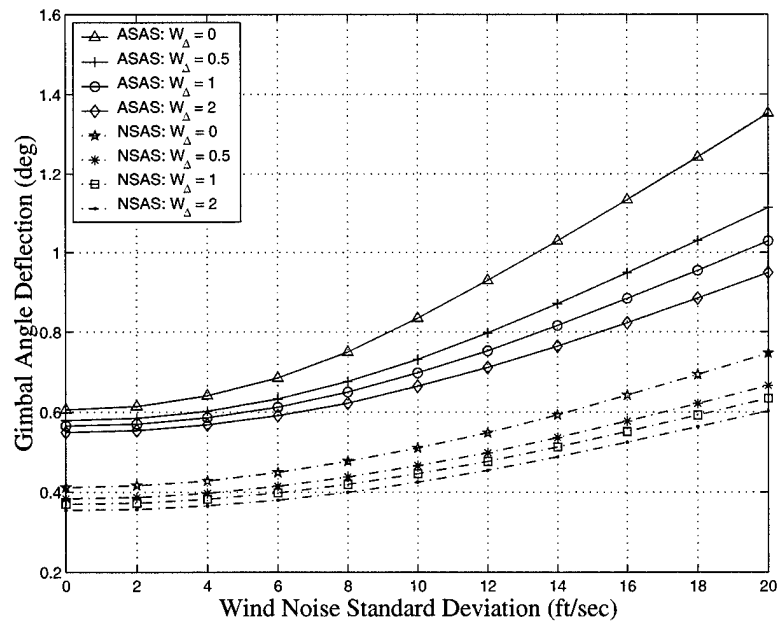
Figure 7-12: Mean Peak Gimbal Angle Sensitivity to W_Δ Weighting
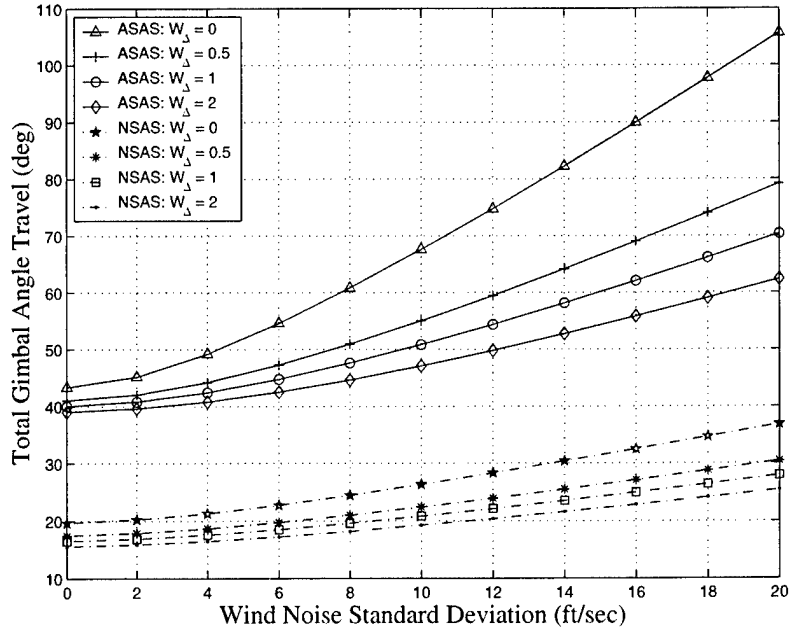
Figure 7-13: Mean Total Gimbal Angle Travel Sensitivity to W_Δ Weighting

α and $Q\alpha$ behavior at low levels of wind uncertainty. However, for these metrics, the $NSAS$ controller shows a much higher sensitivity to increasing amounts of wind uncertainty than the $ASAS$ controller, as shown in Figures 7-9 and 7-10. This may partially be due to the fact that the $ASAS$ controller is using its internal SAS to compensate for the actual winds, while the $NSAS$ controller has no equivalent immediate feedback. Both architectures show low sensitivity to changes in the W_Δ weighting value for these two parameters. This trend is reversed in gimbal activity, as shown in Figures 7-12 and 7-13. Both controllers increase their peak gimbal angle by roughly the same percentage, as wind uncertainty increases, although the $NSAS$ controller's gimbal response is generally 2/3 the size of the $ASAS$ controller's response. In addition, the zero W_Δ weighting in the $ASAS$ controller seems to be significantly more sensitive to increases in wind uncertainty than the equivalent weighting in the $NSAS$ controller. This is most likely due to the internal SAS, whose performance is directly impacted by the uncertainty of the wind disturbance signal, via the α feedback. As can be seen in Figure 7-11, the higher gimbal angles of the $ASAS$ controller more than offset the bending moment reductions created by lower α performance to produce to-

tal bending moments that are approximately 1/3 higher than the equivalent $NSAS$ bending moments, at low wind uncertainty levels. At higher wind uncertainty levels the α and $Q\alpha$ sensitivity of the $NSAS$ controller begins to bring the two controllers to parity in moment performance. Notice the excessive bending moments caused by the zero weighting in the $ASAS$ controller at higher wind uncertainties.

Figures 7-14 and 7-15 show the individual bending moment plots for each controller architecture, which were used to more clearly show the response of each controller.
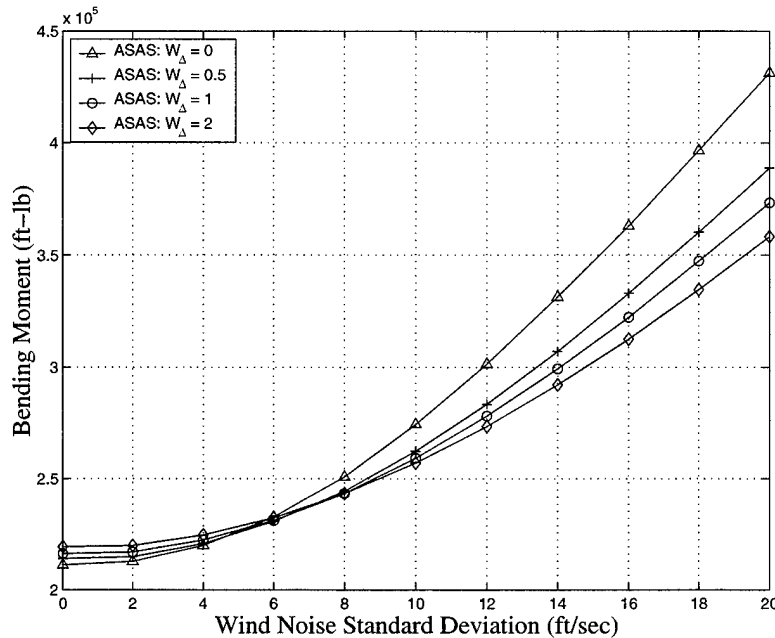
Figure 7-14: Mean Peak Bending Moment Sensitivity of $ASAS$ Controller to W_Δ Weighting

Based on these results and the stated purpose of this research, which was to lower bending moments, the $NSAS$ W_Δ weighting was set to 0 and the $ASAS$ W_Δ weighting was set to 1/2. Recent studies have shown that current LIDAR wind measurement technology is capable of reliably supplying wind disturbance information with accuracies of at least 4 ft/sec or better. Thus, controller performance in this region was of primary importance. However, because of system robustness concerns, the controller performance above 4 ft/sec was also considered. The $NSAS$ controller
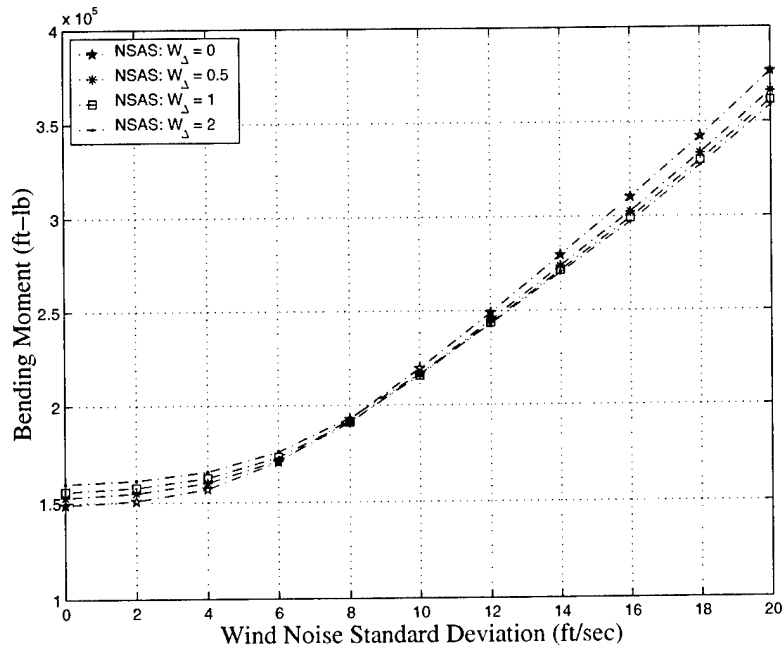
Figure 7-15: Mean Peak Bending Moment Sensitivity of $NSAS$ Controller to W_Δ Weighting

showed very little sensitivity to changes in the W_Δ weighting over the entire wind uncertainty range. The weighting value of 0 showed the best performance below wind uncertainties of 4 ft/sec, and very good performance relative to the other weighting values as uncertainty increased above that point. The $ASAS$ weighting of 0 was deemed unacceptable because of the poor performance at higher wind uncertainties, and the marginal advantage it showed over other non-zero weighting values at low wind uncertainties. The weighting value of 1/2 exhibited nearly the same performance at wind uncertainties below wind noise standard deviations of 4 ft/sec, and very good relative performance beyond that point.

It is important to note that the chosen ratios are only the best of the ratios which were evaluated. It is quite possible that more nearly optimal weighting values could be found if more ratios were investigated. However, for this research, this search was considered sufficient to uncover the significant performance trends caused by W_Δ ratio manipulation.

These plots show that below the wind uncertainty standard deviation level of

98

4 ft/sec, wind uncertainty has virtually no impact on a MPC controller's bending moment performance. As stated previously, this level of wind sensor accuracy is very realistic, given the current state of LIDAR sensor technology development. However, this type of analysis should be retained in the controller development process, as other systems will react differently to disturbance measurement uncertainty, and utilize sensors with varying levels of fidelity.

7.1.6 Prediction Step Size

The prediction step size is the time step size used to propagate the system forward in time. If this variable is set to 0.2 seconds, and the prediction horizon is set to 10, the controller will propagate the system's performance 2 seconds into the future, in 0.2 second increments. Because the controller operates on discrete state-space models, this increment must be used to convert all continuous system models to discrete system models. Shannon's Sampling Theorem dictates that the frequency of sampling must be at least twice the frequency of the fastest unstable pole frequency of the system being controlled [7].

Figure 7-16 shows a plot of the fastest unstable pole in each of the two MPC architectures being considered, at each flight point. The unstable high-frequency aerodynamic pole, which is visible in the $NSAS$ architecture, is stabilized by the inner-loop SAS in the $ASAS$ architecture. Thus the highest frequency unstable pole in the $ASAS$ controller is the low-frequency gravity-induced pole.

Because the inner-loop SAS is controlling α, and not an inertially based parameter, such as pitch angle, this low frequency unstable pole cannot be stabilized. This pole represents the velocity vector, which slowly falls as the vehicle travels along a gravity turn trajectory. Because the controller is attempting to follow this velocity vector with the vehicle's body x-axis vector, the unstable pole results. Since the $ASAS$ architecture has a much lower frequency unstable pole, larger prediction step sizes are possible than with the $NSAS$ architecture. Although this low frequency pole is present in the $NSAS$ architecture, it is overshadowed by the unstable aerodynamic pole, as shown in Figure 7-16.

Using Shannon's Sampling Theorem to calculate the maximum sampling periods possible with these two control architectures produces the curves shown in Figure 7-17. These curves show a theoretical maximum prediction step size of approximately 0.5 seconds for the *NSAS* architecture, and approximately 4 seconds for the *ASAS* architecture.
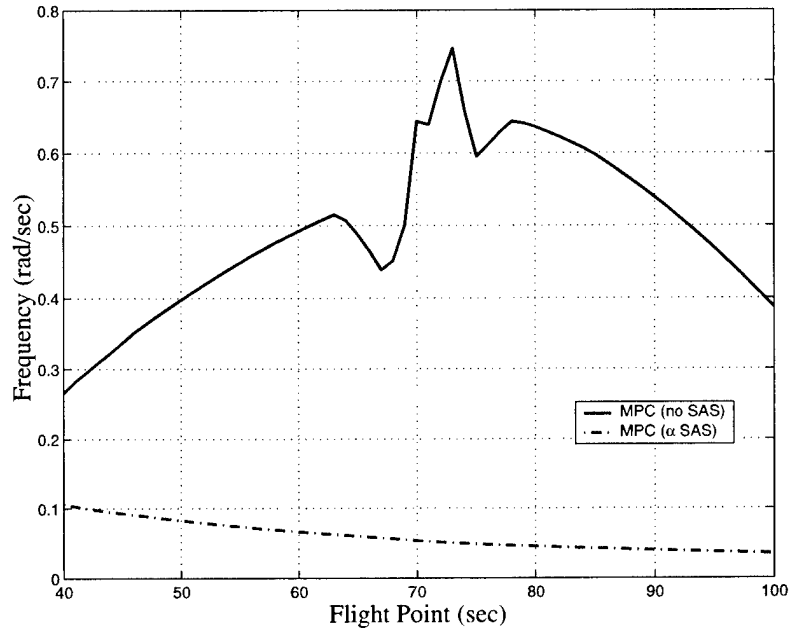
Figure 7-16: Comparison of Highest Frequency Unstable Plant Poles

The available computational power and controller implementation must also be considered when picking the prediction step size. If processor speed is limited, or the optimizer will be operating in real time, a larger step size may be necessary to achieve the desired look-ahead distance with the available computational resources. If more processor power is available, or the controller optimization computations will be completed before flight, a smaller prediction step may be possible. Decreasing sampling time will also allow more accurate disturbance sampling, but will decrease the amount of look-ahead time that any given prediction horizon provides. Likewise, increasing the sampling period will provide a longer look-ahead time for any given prediction horizon and equivalent controller cycle computation load. For flexibility and implementation reasons, possible sampling periods were limited to those that
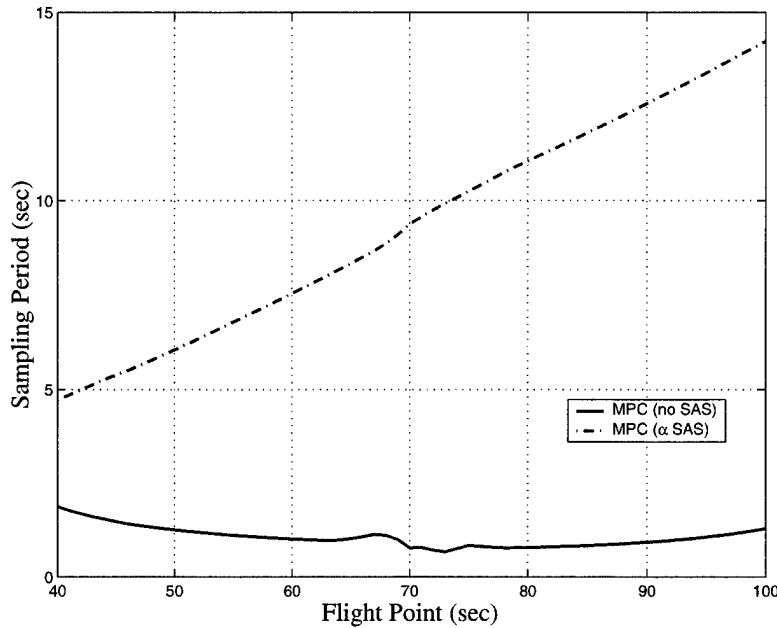
Figure 7-17: Comparison of Maximum Allowable Sampling Periods

could be combined into one second increments, such as 0.1 seconds, 0.2 seconds, 0.5 seconds, etc.

Due to computational limitations, all MPC matrices used in this research were calculated before launch, avoiding many of the computational issues that would exist for a controller seeking to solve the MPC optimization problem in real time. However, an effort was still made to keep the MPC matrix sizes as small as possible while maintaining all significant system performance. To test the sensitivity of each controller architecture to changes in prediction step size, simulation runs were accomplished using the design wind set and a range of prediction step sizes. Each architecture was tested at four step sizes. The *ASAS* controller was tested at 0.1, 0.2, 0.5 and 1.0 second prediction step sizes, while the *NSAS* controller was tested at 0.05, 0.1, 0.2, and 0.5 second step sizes. Step sizes of greater than 1 second were deemed too large to accurately sample the disturbance input, while step sizes of less than 0.05 seconds were deemed too small, as they would require prediction horizons of greater than 80 to meet the previously chosen look-ahead period requirement of 4 seconds. Previously chosen values for the W_Δ weighting values were used. The results of these tests are

shown in Figures 7-18 to 7-22.
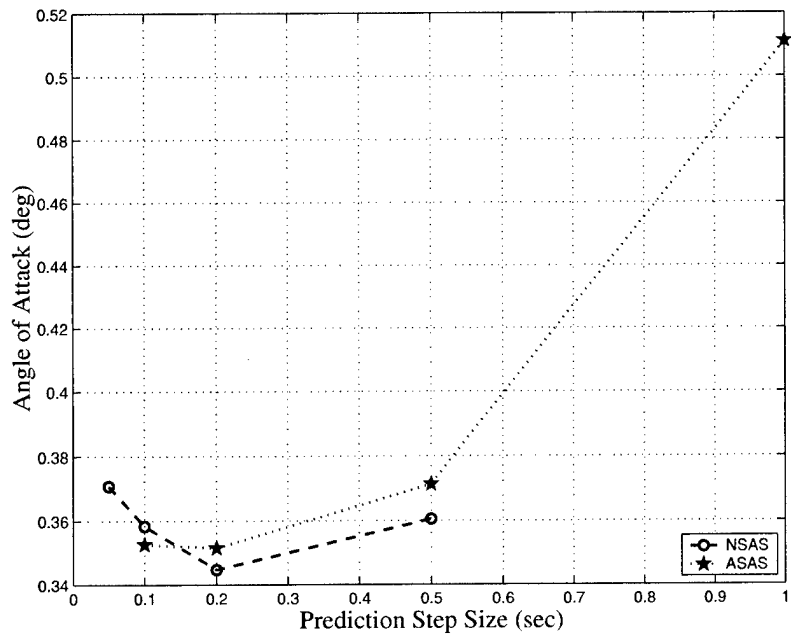
Figure 7-18: Mean Peak Angle of Attack Sensitivity to Prediction Step Size
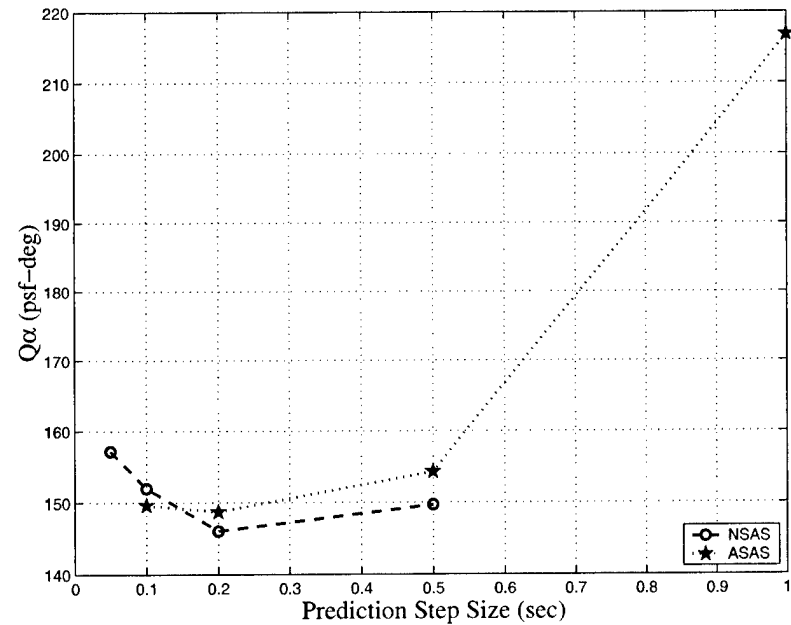
Figure 7-19: Mean Peak $Q\alpha$ Sensitivity to Prediction Step Size
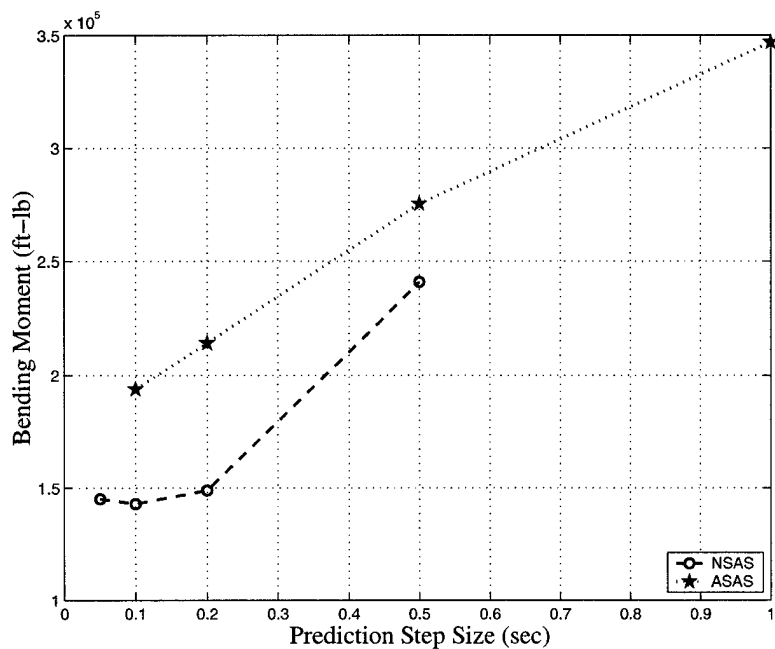
Figure 7-20: Mean Peak Bending Moment Sensitivity to Prediction Step Size

Figure 7-21: Mean Peak Gimbal Angle Sensitivity to Prediction Step Size
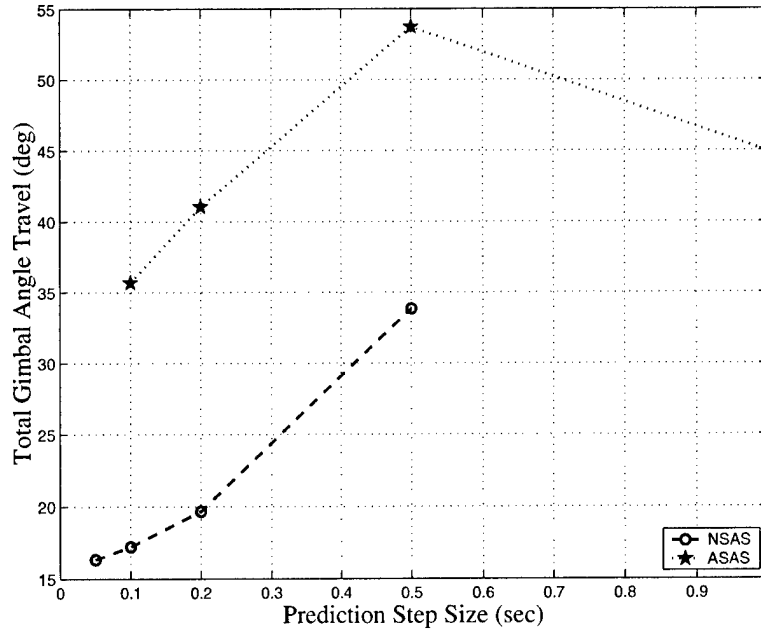
Figure 7-22: Mean Total Gimbal Angle Travel Sensitivity to Prediction Step Size

As in previous analysis, the α and $Q\alpha$ curves are proportionally similar, with the $NSAS$ controller exhibiting a bowl shaped response as the prediction step size is decreased. The increases beyond step sizes of 0.2 seconds are most likely due to a decrease in the gimbal activity at the same step size points. The $ASAS$ controller follows a generally decreasing trend for both of these metrics as step size is decreased, with a slight increase at the smallest step size, for the same reasons as discussed above. The moment curves for both controllers show a general downward trend for the evaluated step sizes. However, as step size continues to decrease, both show progressively smaller reductions in bending moments, with a significant levelling of the $NSAS$ curve below step sizes of 0.2 seconds. The $ASAS$ controller shows similar, though less drastic, levelling behavior. The gimbal activity plots show the same general trend of decreasing gimbal deflection in both controllers as the step size is reduced. Gimbal travel follows this general trend, although the $ASAS$ controller exhibits an interesting increase in gimbal travel at a step size of 0.5 seconds.

Based on these results, the stated goal of lowering bending moments, and the desire to avoid excessive computational loads, the prediction step size for both architectures

104

was set to 0.2 seconds. In each case, the marginal decreases in bending moments beyond this point, as compared to earlier reductions, were not considered significant enough to warrant increasing the prediction horizon with a further doubling from 20 to 40.

7.1.7 J_M Matrix

As previously stated, all MPC matrices used for this research were calculated before flight, due to processor speed constraints. Because of this, the J_M matrix used for all research runs was set equal to an identity matrix of the appropriate size. In general, if a non-identity J_M matrix is desired for processor speed reasons, an evaluation process similar to those described in previous sections of this chapter should be conducted. The performance of multiple J_M matrices would be tested until one was found that yielded suitable time savings while still meeting a given set of controller accuracy requirements.

7.1.8 Controller Rate

Once the prediction step size has been selected, the controller cycle rate can be chosen. The controller rate is the number of times per second that the MPC controller cycles. If this variable is set to 10 Hz, the controller will calculate a new input for the plant 10 times per second. Because the MPC controller implements the first term of the optimal input sequence, the controller rate must be greater than or equal to one over the prediction step size. A controller rate below this limit would result in the controller implementing a control input for longer than it was designed to be used, which would compromise the stability of the controller. Increases in this parameter may be beneficial, but are limited by the available computing power. To determine if any benefits could be gained from increasing the controller rate size beyond the standard rate, a series of trials were completed using the design wind set. Each controller used the settings previously discussed, with controller rates of 5, 10 and 20 Hz for both architectures. Previously chosen values for W_Δ and prediction step size

were used. The results of these trials are shown in Figures 7-23 to 7-27.
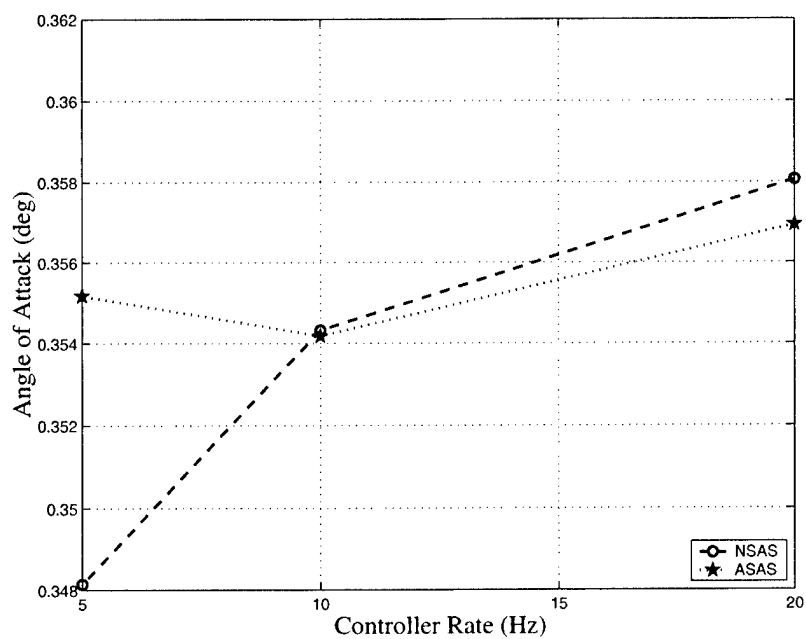
Figure 7-23: Mean Peak Angle of Attack Sensitivity to Controller Rate
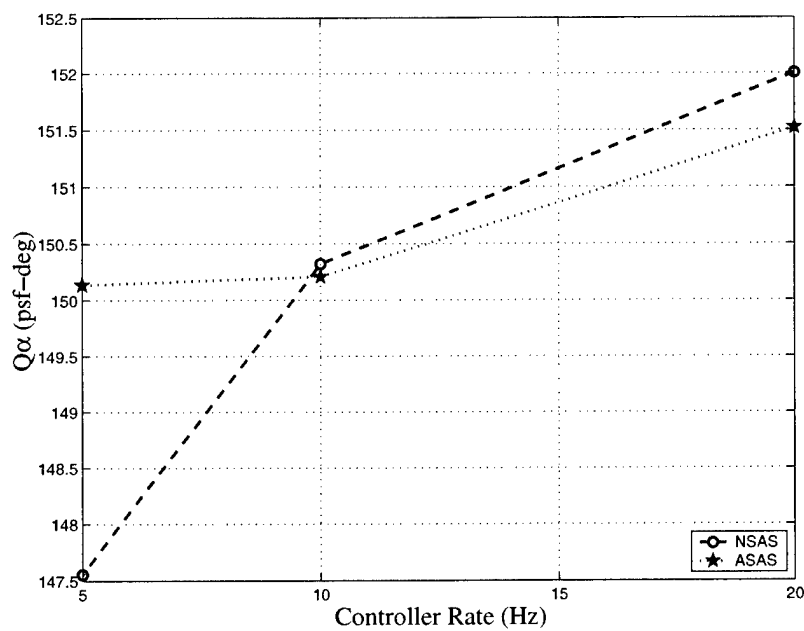
Figure 7-24: Mean Peak $Q\alpha$ Sensitivity to Controller Rate
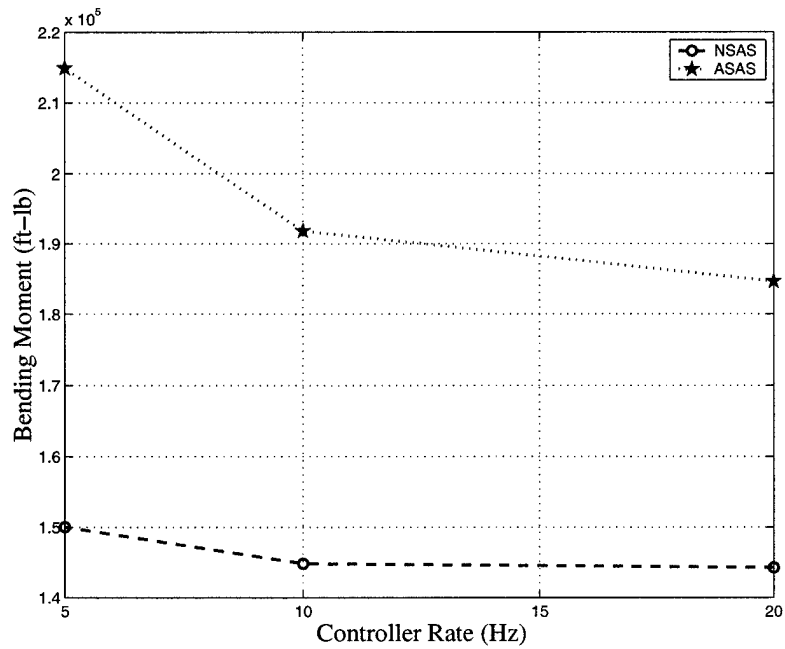
Figure 7-25: Mean Peak Bending Moment Sensitivity to Controller Rate
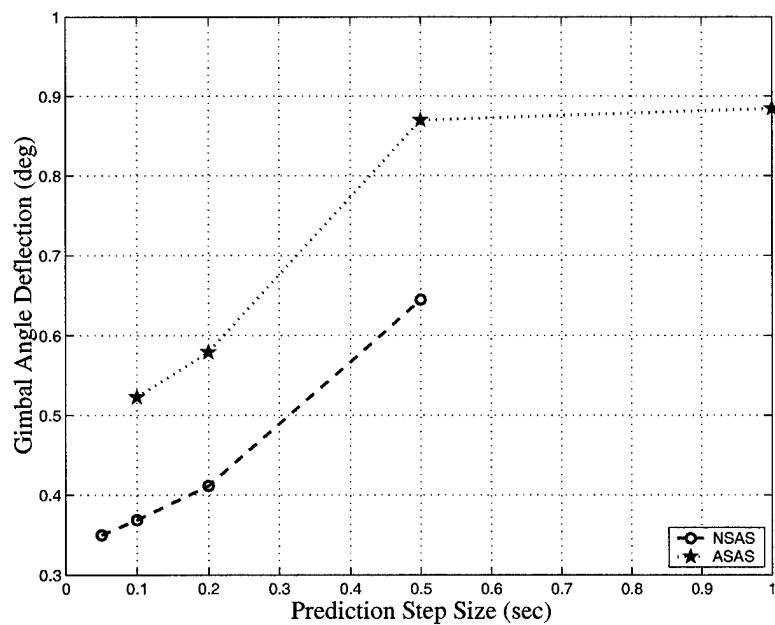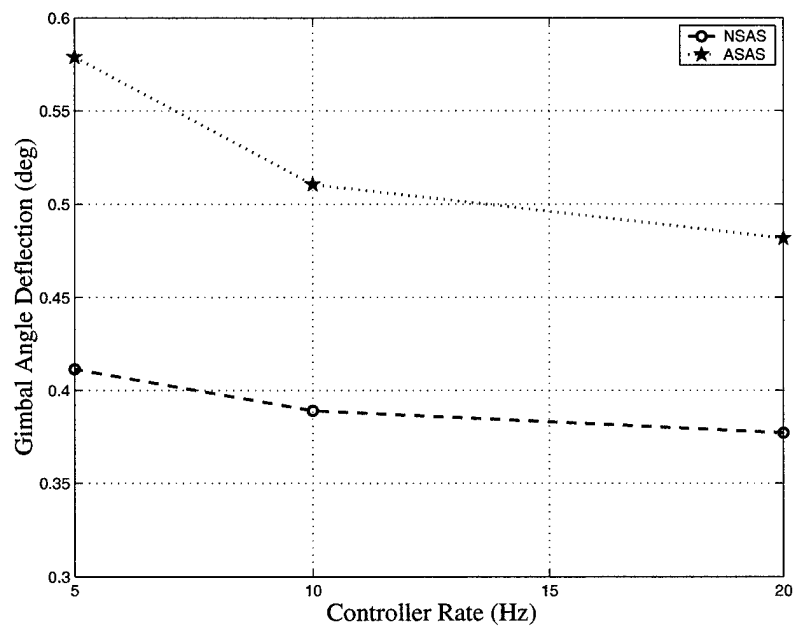

Figure 7-26: Mean Peak Gimbal Angle Sensitivity to Controller Rate
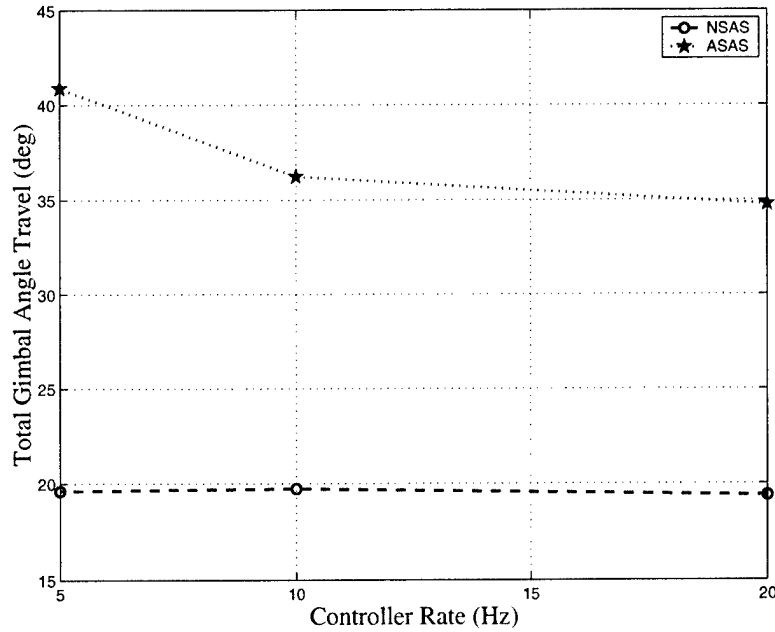
Figure 7-27: Mean Total Gimbal Angle Travel Sensitivity to Controller Rate

These plots show the MPC controller's relatively low level of sensitivity to increases in the controller rate. Both controllers show only marginal decreases in bending moments, gimbal deflection and gimbal travel above rates of 10 Hz. Slight increases in the $NSAS$ α and $Q\alpha$ are evident as controller rate increases. The $ASAS$ controller shows almost no response to increasing controller rate in the α and $Q\alpha$ metrics.

Based on these results and the stated purpose of this research, which was to lower bending moments, the $NSAS$ and $ASAS$ controller rates were both set to 10 Hz. Neither architecture showed benefits above this rate significant enough to justify doubling the rate from 10 to 20 Hz.

7.1.9 Bandwidth Manipulation Sensitivity

When compared to the industry standard baseline controller, the two MPC controllers presented in this chapter show excellent load reduction characteristics, as detailed in Chapter 8. However, these lower α, $Q\alpha$ and bending moment metrics are gained at the price of significantly increased peak gimbal angle deflection and total gimbal

angle travel. Some launch vehicles power their engine gimbals with energy bleeds from the combustion chamber, and can handle large increases in gimbal activity without problem. However, the engine gimbal systems of many modern launch vehicles are powered with open-loop pressure systems, which are only capable of rotating the engines through a preset number of degrees before running out of power. For these vehicles, the high gimbal activity shown by the previously detailed MPC controllers could be problematic.

The ability to trade increased peak α, $Q\alpha$ and bending moment metrics for decreased gimbal activity would be very useful for launch vehicles operating under these limitations. This can be achieved by increasing the weighting value in the input weighting matrix, W_u. This will increase the cost associated with any given gimbal angle movement, which will naturally decrease the control systems ability to respond to wind disturbances, resulting in increased peak α, $Q\alpha$ and bending moment metrics. As mentioned earlier in this section, the important variable is not the actual value of each weighting matrix, but rather the ratio between each set of weighting values. Thus an increase in the weighting value of W_u is equivalent to an appropriate decrease in the weighting value on the output matrix, W_y. Lowering this value causes a decrease in the closed-loop system bandwidth. To investigate the trade-off between gimbal activity and the three metrics of α, $Q\alpha$ and bending moments, $ASAS$ and $NSAS$ controllers with bandwidths of 1, 1.25, 1.5 and 1.75 rad/sec, respectively, were created using the previously chosen variable settings. The performance of these controllers in response to the design wind set is shown in Figures 7-28 to 7-32.

The results reveal several insights into the sensitivity of the MPC controllers. The $ASAS$ controller shows only very minor changes in its performance metrics as the bandwidth is decreased from 1.75 to 1 rad/sec. If the launch vehicle can handle the increased gimbal activity that the $ASAS$ controller produces, this stationary behavior can be an advantage, since the system bandwidth can be lowered substantially without significantly affecting the load reduction performance of the controller. This relative insensitivity to changes in bandwidth is due to the α SAS, which operates at very high cycle rates, resulting in higher gimbal activity than the $NSAS$ controller, regardless

of system bandwidth.
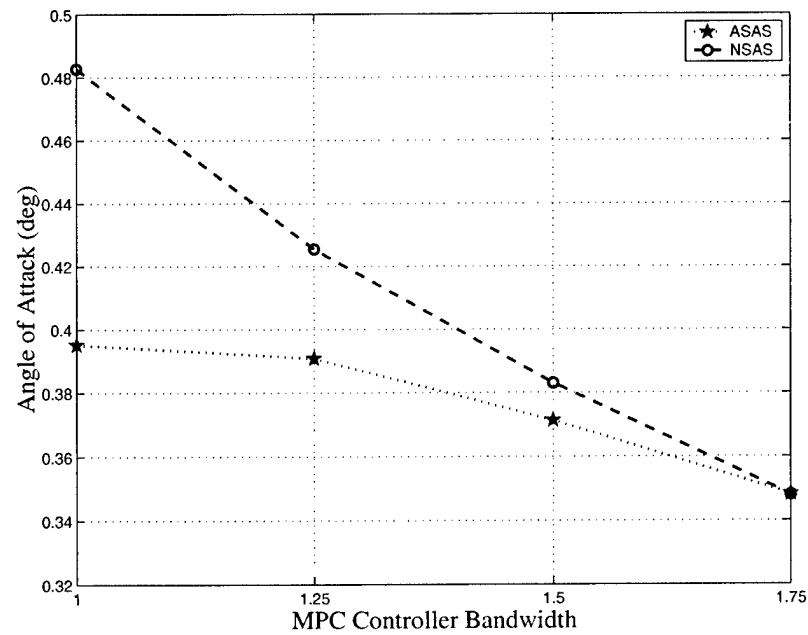
Figure 7-28: Angle of Attack Comparison for Varying Controller Bandwidths
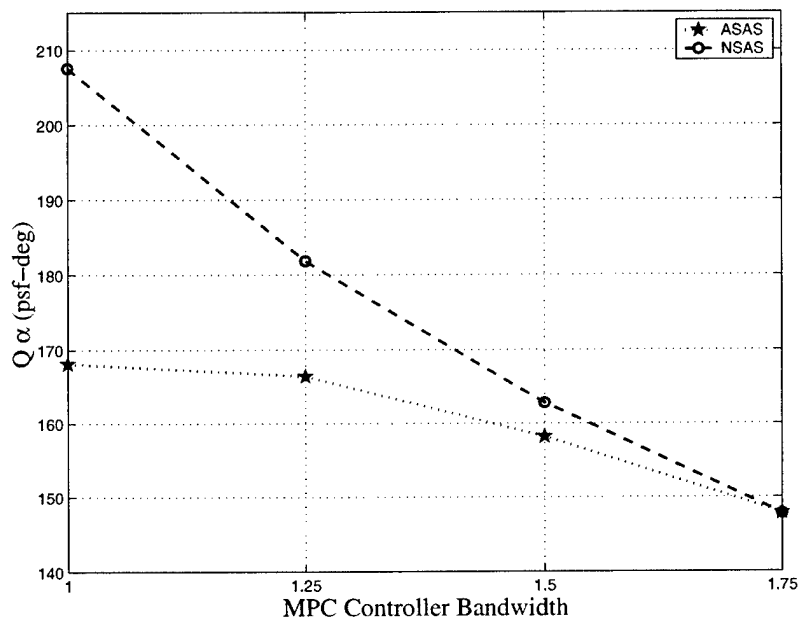
Figure 7-29: $Q\alpha$ Comparison for Varying Controller Bandwidths

110
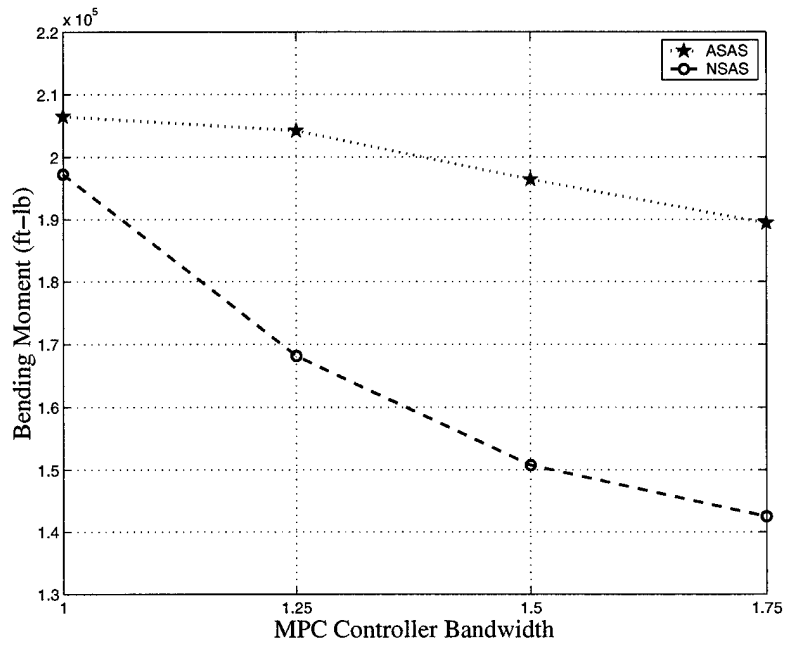
Figure 7-30: Bending Moment Comparison for Varying Controller Bandwidths
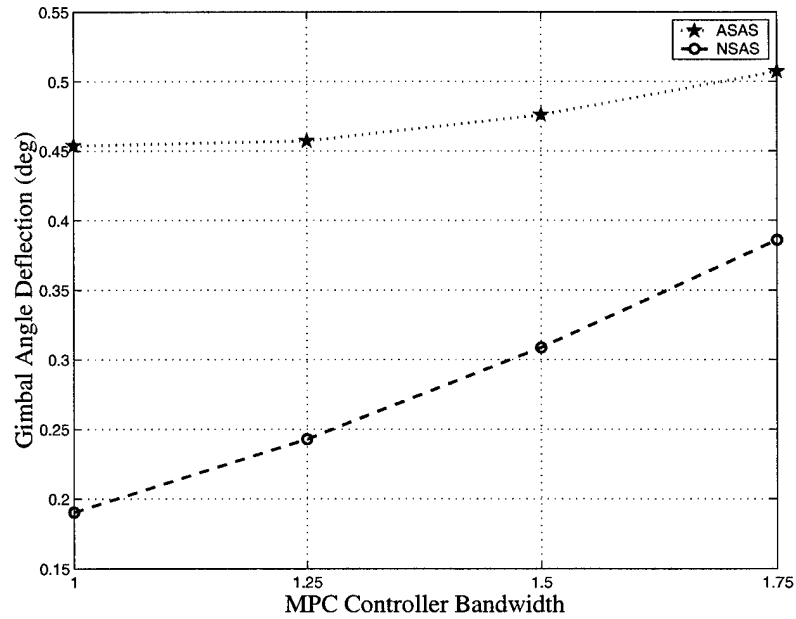
Figure 7-31: Gimbal Angle Deflection Comparison for Varying Controller Bandwidths
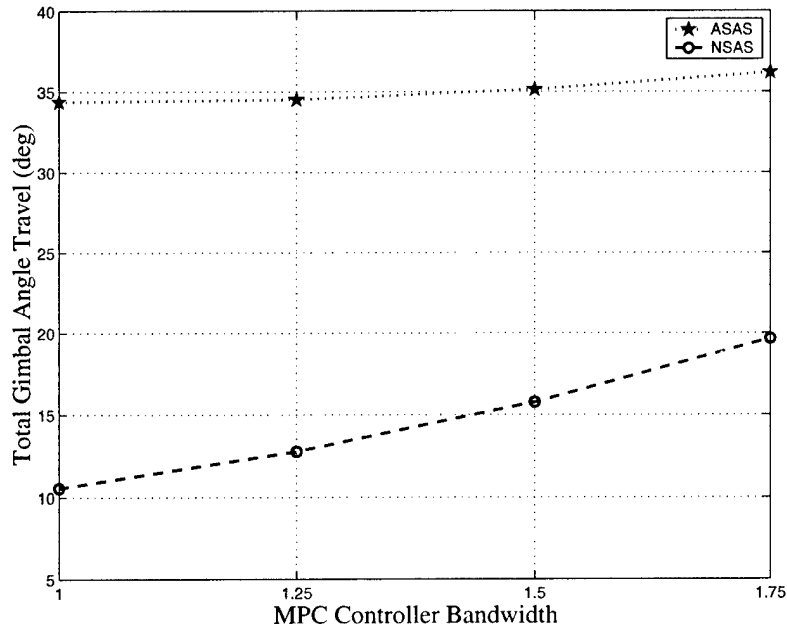
Figure 7-32: Total Gimbal Angle Travel Comparison for Varying Controller Bandwidths

However, if the ability to trade peak bending moment or $Q\alpha$ performance in exchange for decreased gimbal activity is desired, the $NSAS$ controller must be used. As shown in the previous figures, the $NSAS$ controller metrics vary significantly as the system bandwidth is lowered from 1.75 to 1 rad/sec. These changes result in an approximate 50% decrease in the gimbal angle deflection and total gimbal angle travel, in exchange for an approximate 35-40% increase in α, $Q\alpha$ and bending moments over this range. This is a very important capability if special consideration must be taken for a given launch vehicle's engine gimbal movement limitations.

The K-1 engine gimbals are powered by an accumulator drawing power directly from the vehicle and are not subject to movement limitations which approach the levels shown in this thesis. Because of this, the K-1 can handle large increases in engine gimbal movement without any significant penalty to the vehicle itself. Thus, pursuant to the stated goal of this research, which is to minimize bending moments during ascent, the previously chosen closed-loop system bandwidth of 1.75 rad/sec will be used during the remainder of this research.

7.1.10 Controller Variable Summary

The results of the controller tuning process are summarized in the Table 7.2:

MPC Controller Type	No SAS	α SAS
Closed-Loop System Bandwidth (rad/sec)	1.75	1.75
Look Ahead Period (sec)	4	4
Prediction Horizon (-)	20	20
W_y Weighting (-)	Adjusted for Bandwidth	Adjusted for Bandwidth
W_u Weighting (-)	1	1
W_Δ Weighting (-)	0	0.5
Prediction Step Size (sec)	0.2	0.2
J_M Matrix	Identity	Identity
Controller Rate (Hz)	10	10

Table 7.2: MPC Controller Parameter Setting Summary

7.2 Wind Processing Analysis

As described in Chapter 5, the wind profiles used for this research contain wind information in approximately 80 ft increments. Current LIDAR wind sensors are capable of discerning wind information in increments of between approximately 35 to 245 ft. From this information, it is possible to pre-process the wind data in a number of different fashions, including higher order polynomial fits, varying sampling increments or moving average smoothing. In this case, moving average smoothing was the method of choice because it retains a relatively high amount of accuracy, compared to sampling, as the averaging distance is increased. Sampling at various increments and linearly interpolating between data points was also investigated and found to be deficient, due to poor accuracy retention as the increment distance was increased. Polynomial fits were unacceptable because they could not accurately represent wind profiles at very high levels of fidelity. To determine whether an advantage might be

gained from some level of wind processing, as well as to investigate each controllers response to less and less accurate wind information, both MPC architectures were flown against the design wind set, using forward and backward moving average distances of 0, 1, 2, 4, 8, 16, 24, and 32 data points, respectively. The spacing between these points ranges from 0 ft to approximately 2600 ft in each direction, hereafter called the ± distance.

Figures 7-48 to 7-52 show an interesting set of trades between gimbal activity and bending moments for both MPC controllers. At certain points along the curves, favorable trades exist which could be used to modulate the controller's performance to compensate for individual launch vehicle limitations in gimbal movement capability. As in the bandwidth manipulation example, the $ASAS$ controller does not show a significant gimbal activity decrease in response to changes in sensed wind profile smoothing, relative to the much larger increases in the α and $Q\alpha$ metrics. This results in bending moment metrics which increase at unacceptable rates in response to small or negligible decreases in gimbal activity. As noted before, this is because of the internal α SAS, which operates separately at nearly real-time rates, in response to the actual, unprocessed wind profile. Thus, the gimbal activity is relatively unresponsive to smoothing of the sensed disturbance vector, as this vector is loaded into the MPC controller, but not the α SAS.

The $NSAS$ controller shows more gimbal sensitivity to the smoothing of the disturbance vector. Although the bending moment curve is continually increasing, there are points where the percentage increase in bending moments is significantly outweighed by the percentage decrease in gimbal angle deflection and total gimbal angle travel. In a vehicle with limited gimbal movement capability, this option can prove important. An example smoothing average ± distance of 330 ft shows an approximately 3% increase in bending moments in exchange for an approximate decrease in gimbal angle deflection and total gimbal angle travel of 20%.

In light of these results, the stated purpose of this research, and the K-1's robust ability to absorb increases in commanded total gimbal angle travel, all wind disturbance profiles will be loaded into the MPC controllers at the highest level of fidelity

114

possible, which is in approximately 80 ft increments in this case.
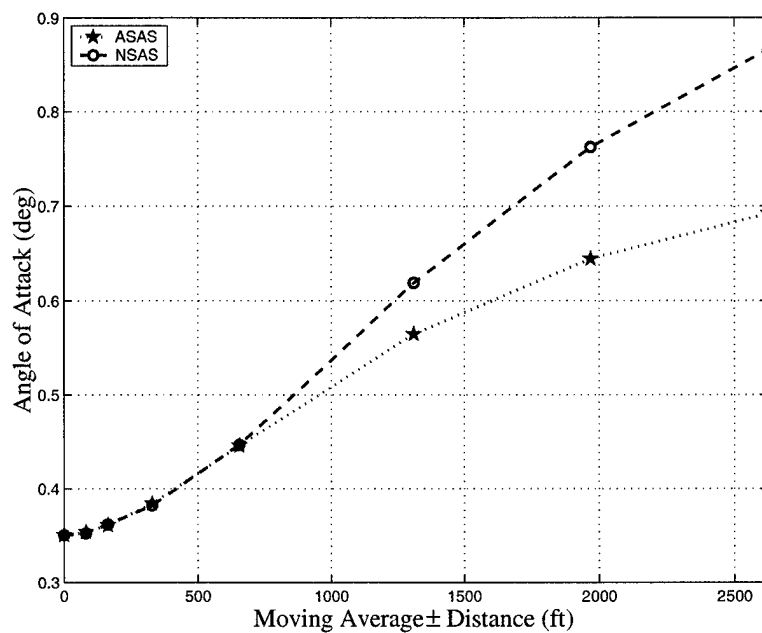
Figure 7-33: Angle of Attack Comparison for Varying Wind Speed Moving Average ± Distances
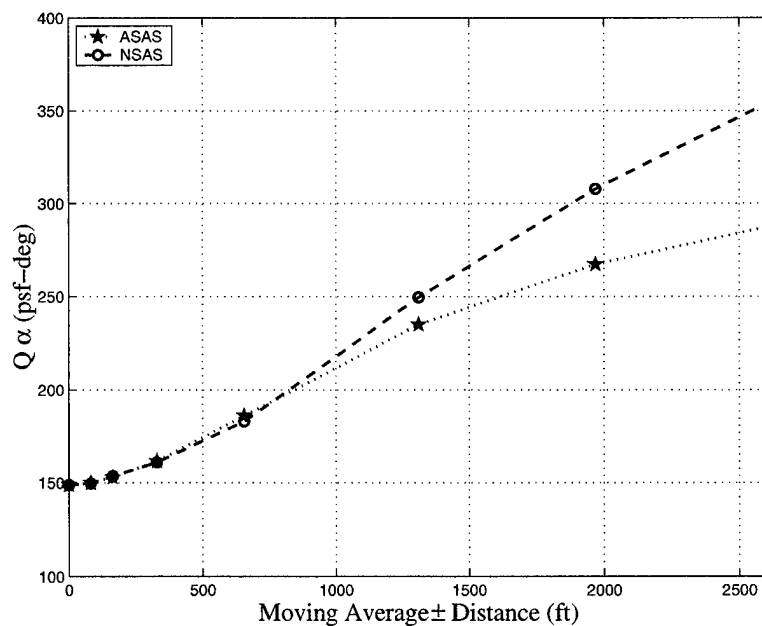
Figure 7-34: $Q\alpha$ Comparison for Varying Wind Speed Moving Average ± Distances
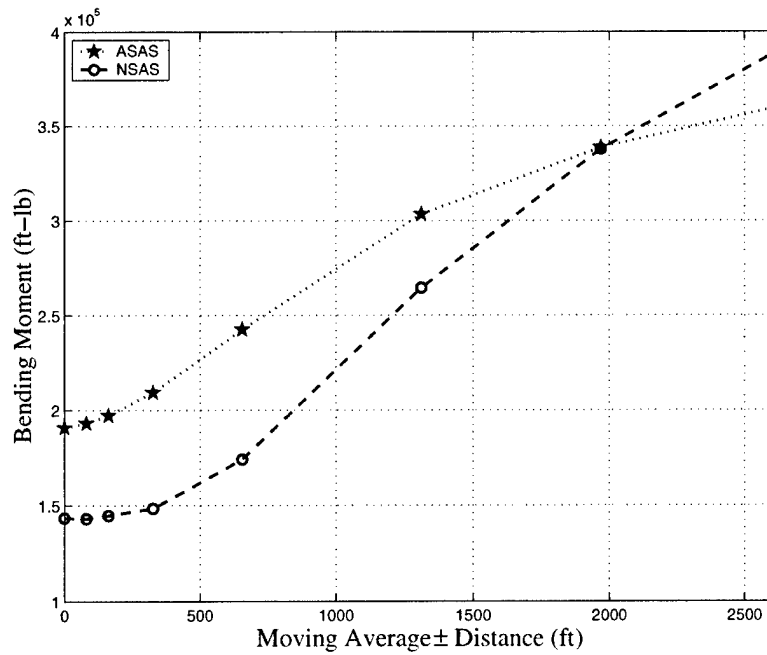
Figure 7-35: Bending Moment Comparison for Varying Wind Speed Moving Average ± Distances
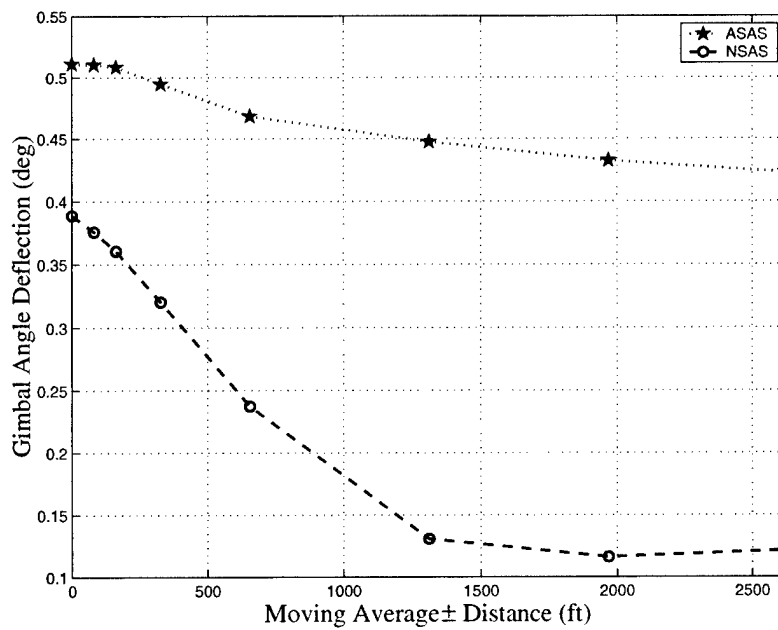
Figure 7-36: Gimbal Angle Deflection Comparison for Varying Wind Speed Moving Average ± Distances
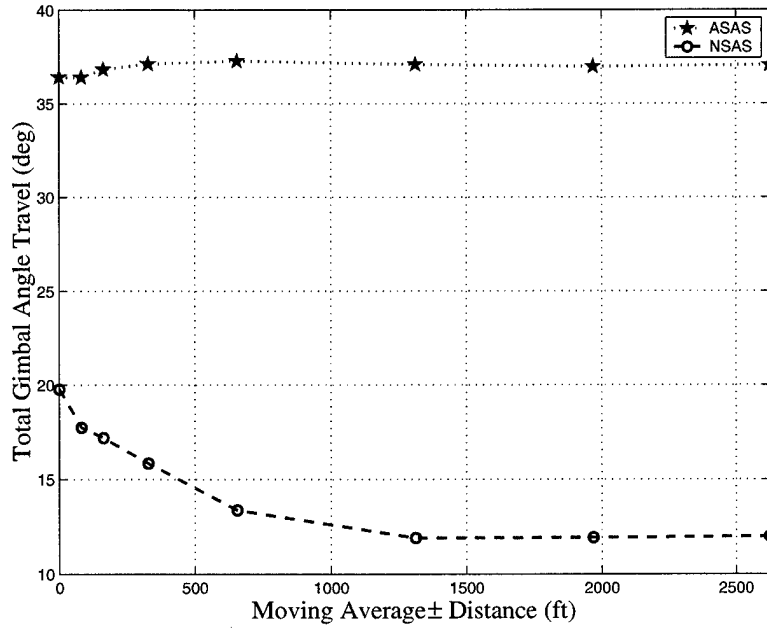
Figure 7-37: Total Gimbal Angle Travel Comparison for Varying Wind Speed Moving Average ± Distances

7.3 Controller Stability Analysis

The stability analysis of each MPC controller was complicated by the presence of a low frequency unstable pole, as noted previously. This pole was created by commanding the vehicle to follow a variable trajectory that was not inertially based, in this case a given α trajectory. Commanding the vehicle to follow a variable such as pitch would eliminate this pole. However, because of the load-relief focus of this research, it was necessary to command an α trajectory, as opposed to a pitch or flight-path angle trajectory. This mode diverged very slowly, and was very unresponsive to gain manipulation. To calculate the gain and phase margin of the system, a dedicated SIMULINK model was used, much like the model used to calculate closed-loop system bandwidth. To calculate the gain margin, the input command from the controller to the plant was multiplied by larger and larger gains, until one of the system poles, aside from the small pole due to α following, crossed the $j\omega$ axis. This value was then

converted to find the gain margin of the system in dB. The phase shift was calculated in the same manner. The input command was affected by larger and larger time delays until the first stable pole was driven into the right hand plane. The delay was implemented by a 2nd order pade linear approximation of the actual non-linear delay, $e^{-j\omega\tau}$. This time delay was then used with the magnitude of the pole crossover frequency to calculate the phase margin. This process was repeated at each flight point for both the $ASAS$ and $NSAS$ controllers. A picture of the SIMULINK model used to calculate the gain and phase margins for the $NSAS$ controller is included below.
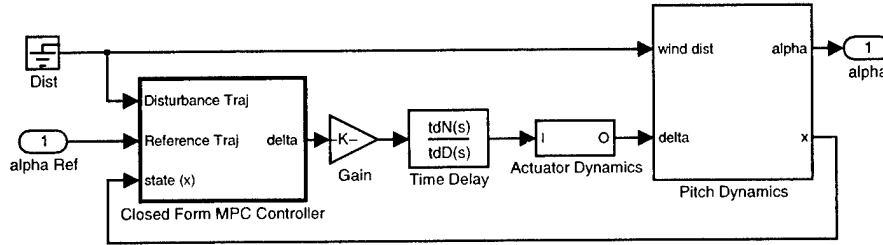
Figure 7-38: Stability Analysis Model for $NSAS$ Controller

Although not explicitly included in the design process, gain and phase margin requirements of 6 dB and 30 deg, respectively, were implicitly imposed on the resulting systems. With an average gain margin of more than 18 dB and an average phase margin of more than 55 deg, both MPC controllers far exceeded the stability requirements at all flight points. Figures 7-39 and 7-40 show plots of sample pole tracks as gain and time delay are increased, respectively. The gain and phase margins of both controllers are shown in Figures 7-41 and 7-42.

A bode plot of each system at each flight point is shown in Figures 7-43 and 7-44. Notice the 1.75 rad/sec frequency, where all magnitude responses converge at the -3 dB line. A DC gain of less than one is visible in the $ASAS$ bode plot. This will result in slight tracking errors, which must be compensated for during flight. The solution to this problem is discussed in the Controller Implementation section.
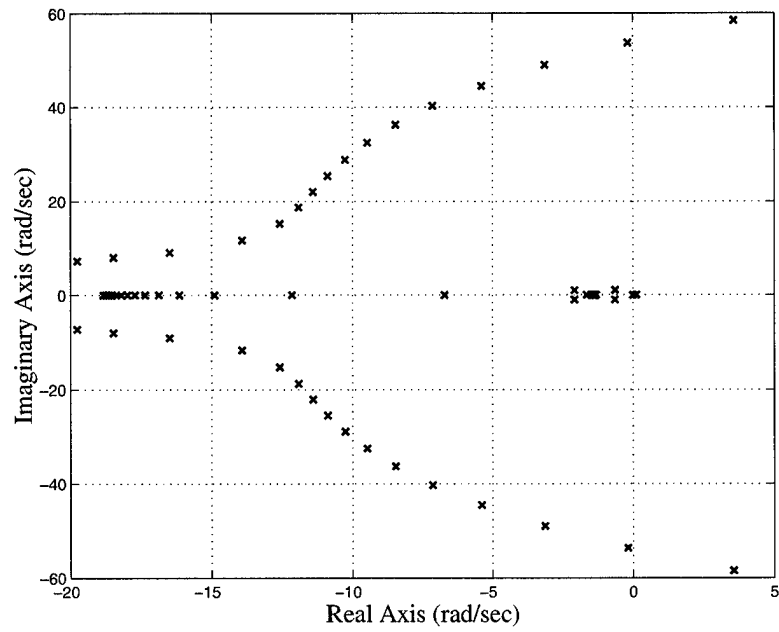
Figure 7-39: Pole Positions for Increasing Gain Values
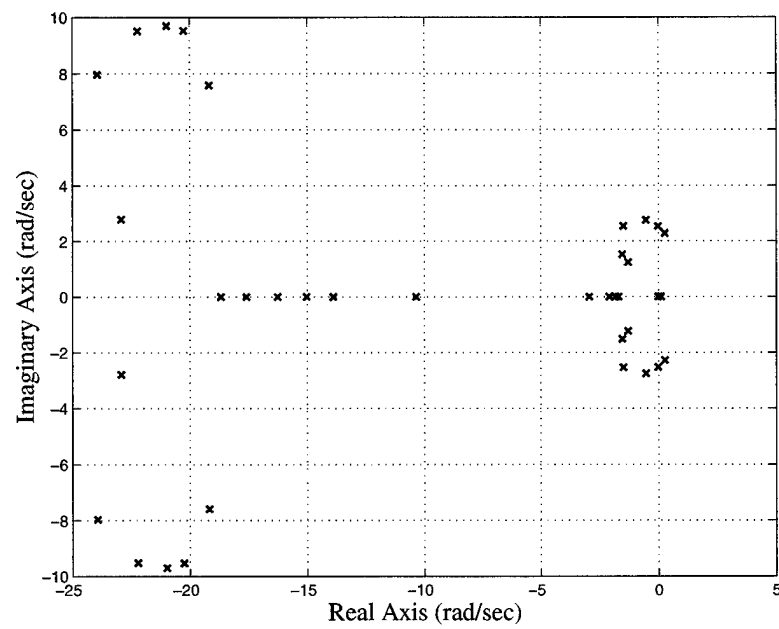
Figure 7-40: Pole Positions for Increasing Time Delay Values
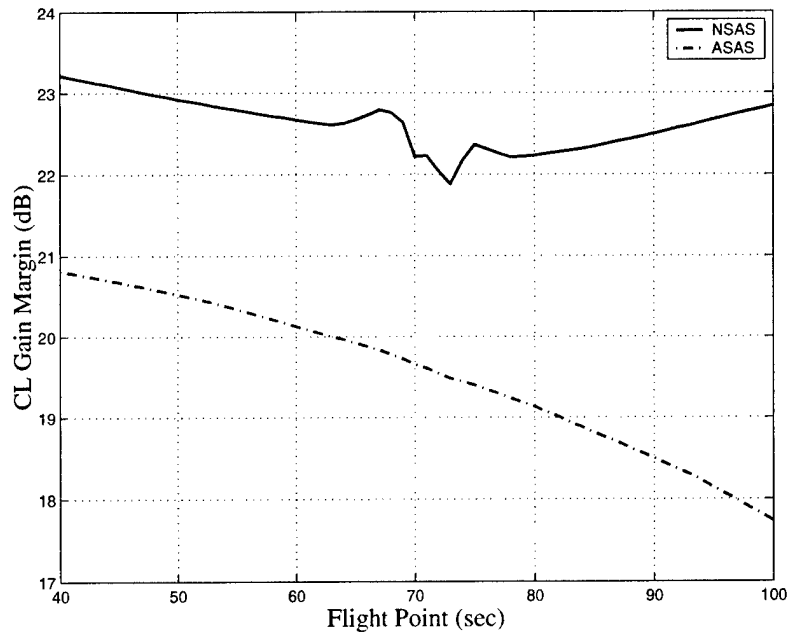
Figure 7-41: Gain Margin of MPC Controllers at each Flight Point
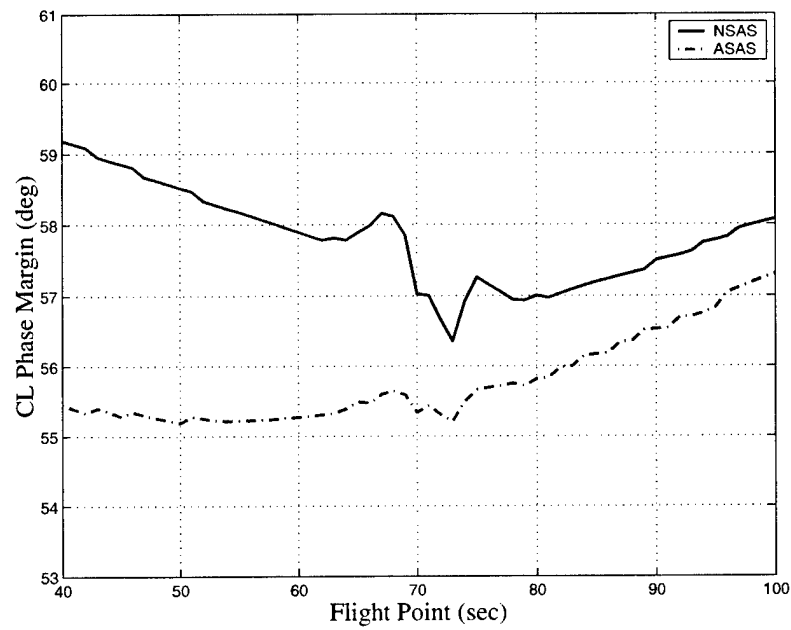
Figure 7-42: Phase Margin of MPC Controllers at each Flight Point
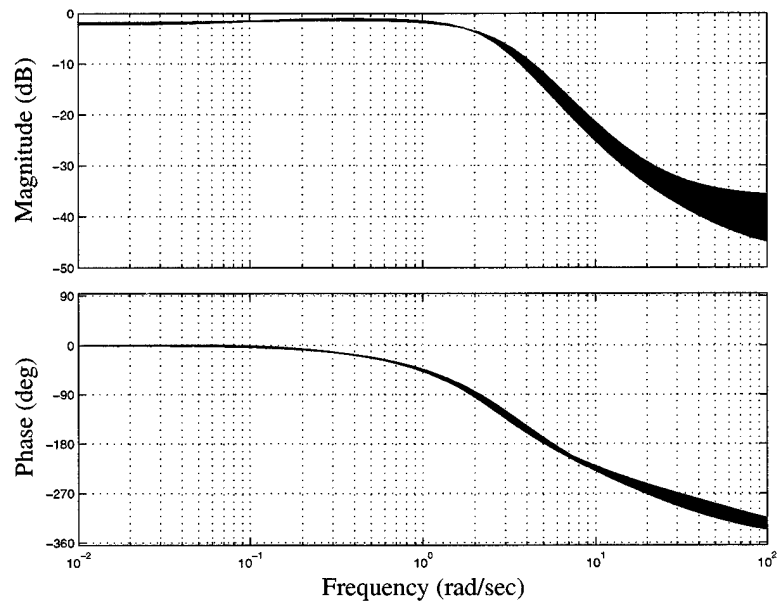
Figure 7-43: Closed Loop Bode Plot of $ASAS$ Controller at each Flight Point
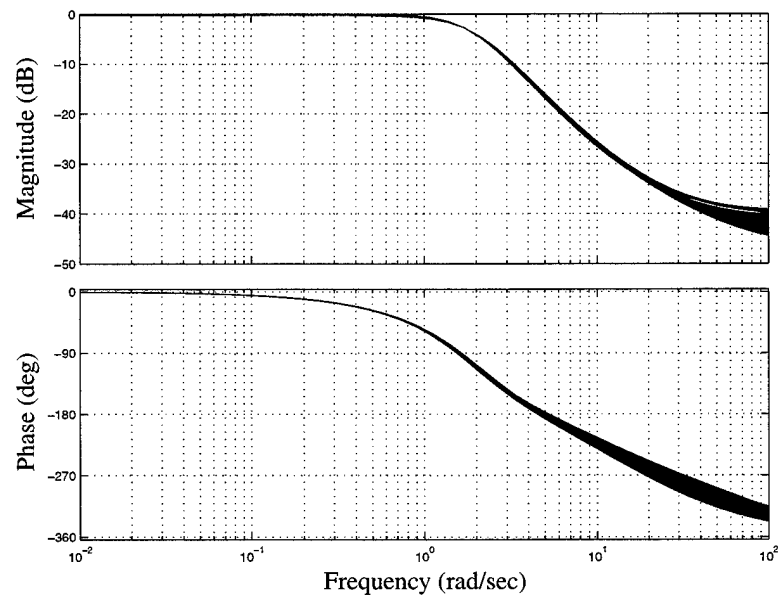
Figure 7-44: Closed Loop Bode Plot of $NSAS$ Controller at each Flight Point
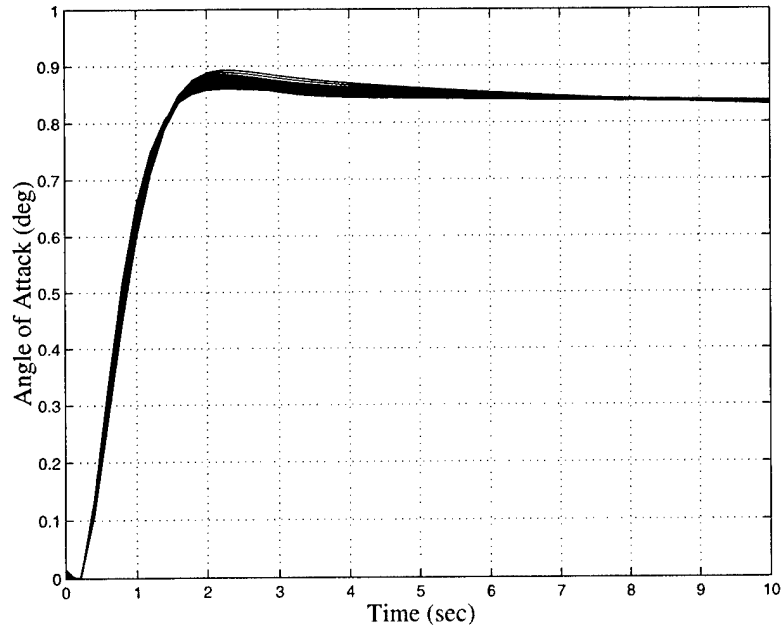
Figure 7-45: Step Response of $ASAS$ Controller at each Flight Point
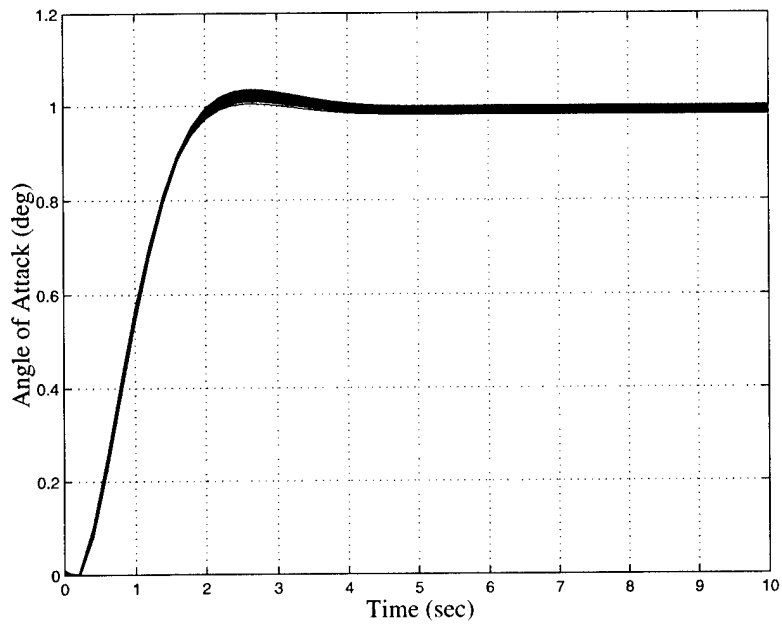
Figure 7-46: Step Response of $NSAS$ Controller at each Flight Point

While no explicit system performance requirements were imposed during the design process, the performance of each controller at each flight point was evaluated

to ensure that large performance gaps did not exist between each controller's performance and standard "acceptable" performance criteria. Figures 7-45 and 7-46 show plots of the step response of each controller, respectively, at each flight point. The *ASAS* controller average percent overshoot was less than 5%, while the average rise time was about 1.4 seconds. The *NSAS* controller average percent overshoot was less than 5%, while the average rise time was about 1.6 seconds. All metrics were well within acceptable limits. The steady-state error of the *ASAS* controller is clearly visible, and a very small steady state error is visible for most of the *NSAS* controller step responses.

7.4 Controller Implementation

Chapter 3 detailed the construction of the MPC matrices: K_r, K_v, K_d, K_u, K_d, K_t and Kdu. Using these matrices, the optimal input parameter sequence, $\overline{z}^*(k)$, can be calculated. Because the J_M matrix used for this research was an identity matrix, this vector is equal to the optimal input control change sequence, $\overline{\Delta u}^*(k)$. This equation is shown below:

$$\overline{\Delta u}^*(k) = -K_{du}^{-1}[\overline{r}^T(k)K_r + \overline{v}^T(k)K_v + \overline{d}^T(k)K_d +$$
$$u^T(k-1)K_u + \overline{u}_T^T(k)K_T + x^T(k)K_x]^T \quad (7.1)$$

This equation can be rewritten as:

$$\overline{\Delta u}^*(k) = -K_{du}^{-1}[\overline{r}^T(k)K_r]^T - K_{du}^{-1}[\overline{v}^T(k)K_v]^T - K_{du}^{-1}[\overline{d}^T(k)K_d]^T -$$
$$K_{du}^{-1}[u^T(k-1)K_u]^T - K_{du}^{-1}[\overline{u}_T^T(k)K_T]^T - K_{du}^{-1}[x^T(k)K_x]^T \quad (7.2)$$

For this research, only measured disturbances were considered, thus eliminating the K_d matrix. The remaining five terms were reorganized to create the following

matrices.

$$Kr = -K_{du}^{-1}K_r^T \qquad (7.3)$$

$$Kv = -K_{du}^{-1}K_v^T \qquad (7.4)$$

$$Ku = -K_{du}^{-1}K_u^T \qquad (7.5)$$

$$Kt = -K_{du}^{-1}K_T^T \qquad (7.6)$$

$$Kx = -K_{du}^{-1}K_x^T \qquad (7.7)$$

Using these newly defined terms, Equation 7.2 can be rewritten to yield the optimal control sequence, as shown below.

$$\overline{\Delta u}^*(k) = Kr * \bar{r}(k) + Kv * \bar{v}(k) + Ku * u(k-1) + Kt * \bar{u}_T(k) + Kx * x(k) \qquad (7.8)$$

Because only the first term of any given optimal control sequence calculated by the MPC controller is implemented, only the first row of each of these matrices must be stored at each flight point. During flight, a linear interpolation is completed between the appropriate flight points to determine the correct interpolated K vector. Once each of these vectors is calculated, only a simple series of vector multiplications and summations is required to calculate the first term of the optimal control sequence. This allows extremely fast execution times with very low computational power requirements. Figure 7-47 shows a schematic of the closed-form SIMULINK MPC controller setup used by both MPC architectures.

The summing junction immediately before the controller output adds every successive Δu term to the last commanded input, creating the next commanded input. To implement this in a SIMULINK environment, the unit delay is required. Each of the five K vectors are linearly interpolated in the blocks shown. The disturbance trajectory is a vector of the anticipated wind disturbance magnitudes at each prediction step from the present time to the prediction horizon. These vectors are gathered
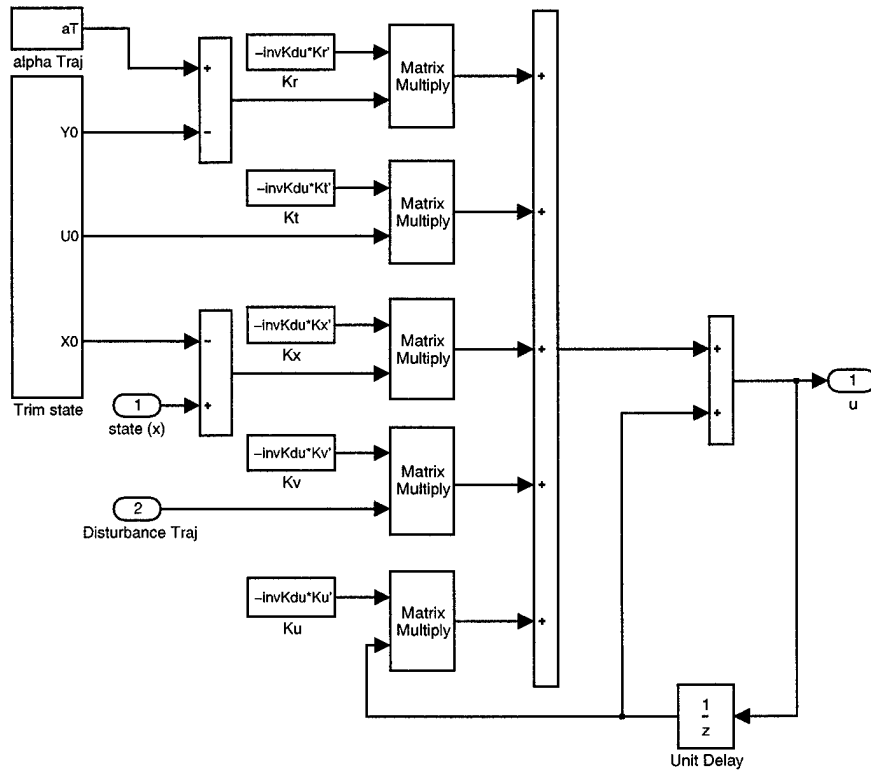
Figure 7-47: MPC Controller SIMULINK Implementation

from a look-up table relating altitude to wind disturbance magnitude. Because this analysis is completed only in the pitch plane, the wind magnitude information is in the inertial X-Z plane. The anticipated altitude of the vehicle is calculated using a simple kinematics equation and the vehicle's present position, velocity and acceleration. The state vector is the current state from the vehicle, in this case the current $[\dot{\theta} \ \theta \ u \ w]$, where u and w are the body velocities in the X and Z inertial directions respectively, as defined in Chapter 2. Y0, U0 and X0 represent the trim output, input and state values that were present during the linearization process. X0 is the current trim state, while Y0 and U0 are projections of the trim values forward in time to the projection horizon. The vector aT represents the commanded output, in this case some commanded α trajectory. All sampling times in the input ports, output ports and digital clocks are set to the controller rate, while the prediction step is used in all blocks which are used to find predicted values from look-up tables, such as the trim

125

states and α trajectory.

7.5 Controller Bias Processing Analysis

Due to the MPC implementation method, both controllers exhibited tracking errors during flight. This caused the controller to drift slightly during flight, usually resulting in an α trajectory which contained a 1/2 to 1/4 degree error, in relation to the commanded zero α trajectory. The solution to this problem was to run the simulation twice. The first run would serve as an initialization run, while the second run would be considered the actual flight. During the load relief portion of the initialization run, the commanded α trajectory would be set to zero, and the resulting α trajectory would be recorded. During the second simulation run, the commanded α trajectory would be set equal to some filtered version of the negative of the previously recorded actual α trajectory. This would effectively cancel out all significant drift that the controller may have exhibited during the initialization run. This correction/command signal is represented by the aT vector shown in Figure 7-47.

Because the recorded α signal contains significant high frequency content, it may be advantageous to process this signal before using it as the commanded α trajectory. As previously stated in the wind processing analysis section, the processing method of choice is to smooth the alpha bias signal using a moving average. To determine how both MPC controllers respond to different amounts of smoothing, both MPC simulations were flown against the design wind set with moving average $\pm$ distances of 0, 2, 4, 8, 16, 24, 32, 40, 48, 56 and 64 data points, respectively. These equate to $\pm$ time spans ranging from 0 to 6.4 seconds. The results are shown in Figures 7-48 to 7-52.

These figures show an interesting set of metric curves from the MPC controllers in response to different amounts of α bias signal smoothing. In this case, it seems possible to conduct an acceptable trade between gimbal activity and bending moments for both controllers. The *ASAS* controller shows a significant response to smoothing of this signal, as it is a primary driver of the gimbal activity produced by the α SAS.
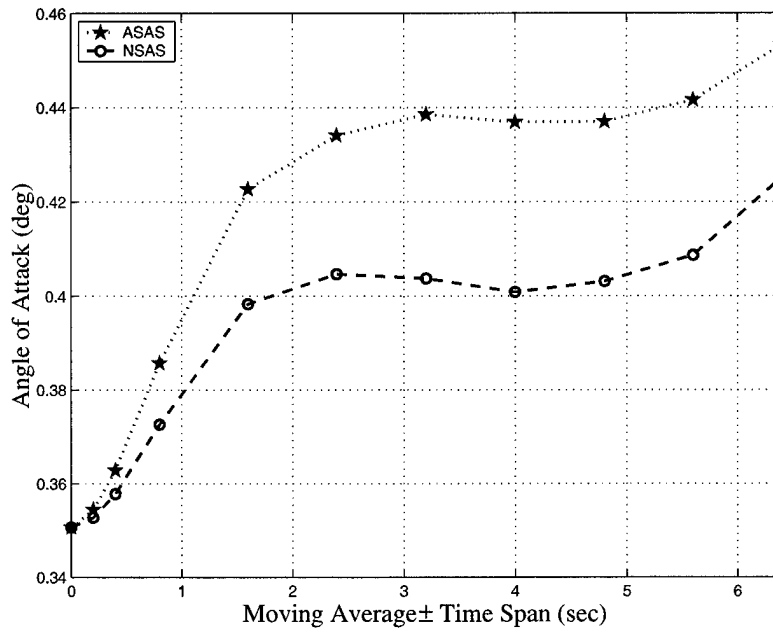
Figure 7-48: Angle of Attack Comparison for Varying Bias Moving Average ± Time Spans
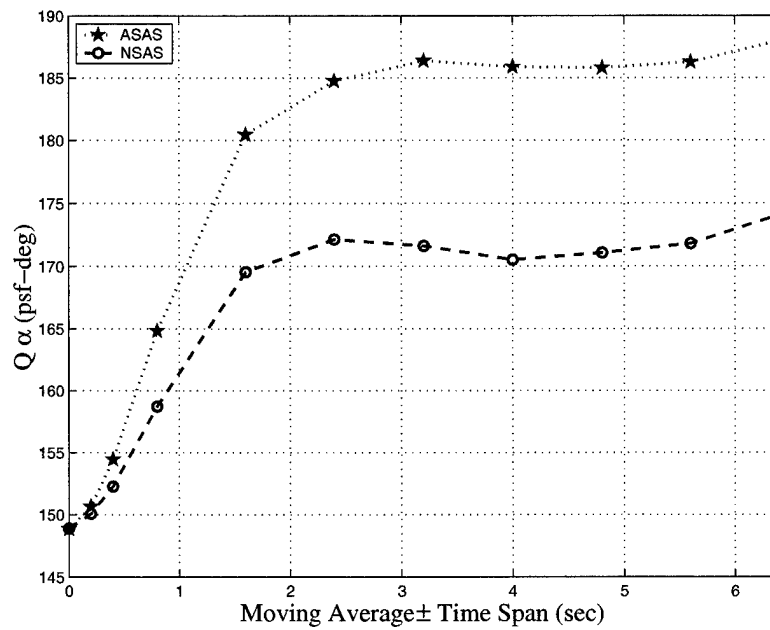
Figure 7-49: $Q\alpha$ Comparison for Varying Bias Moving Average ± Time Spans
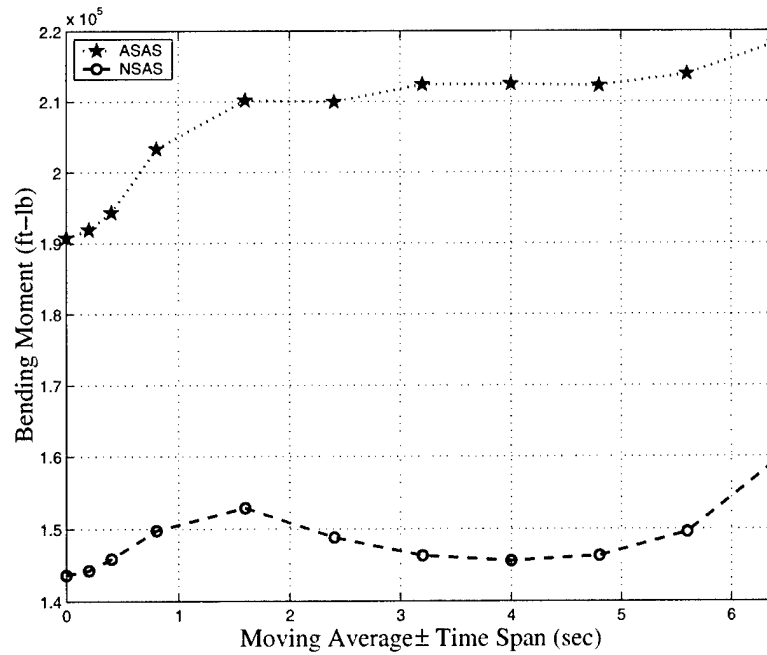
Figure 7-50: Bending Moment Comparison for Varying Bias Moving Average ± Time Spans
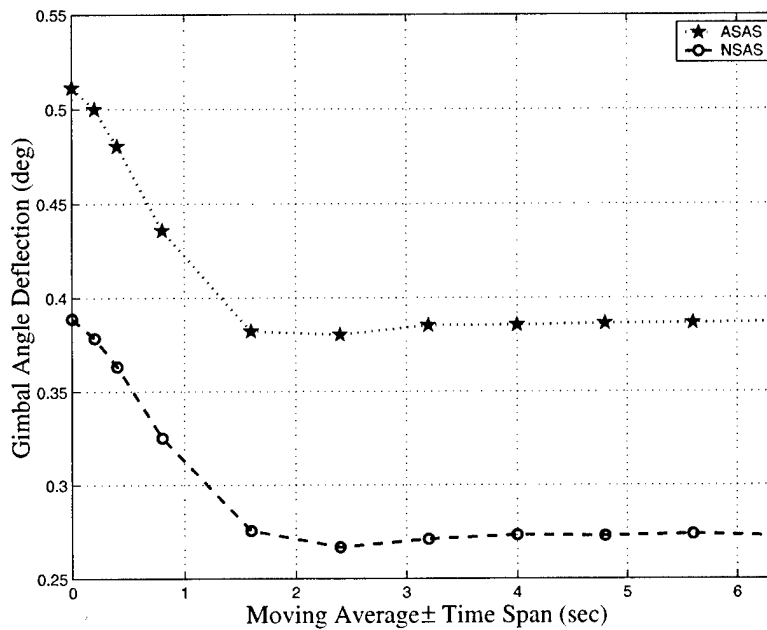
Figure 7-51: Gimbal Angle Deflection Comparison for Varying Bias Moving Average ± Time Spans
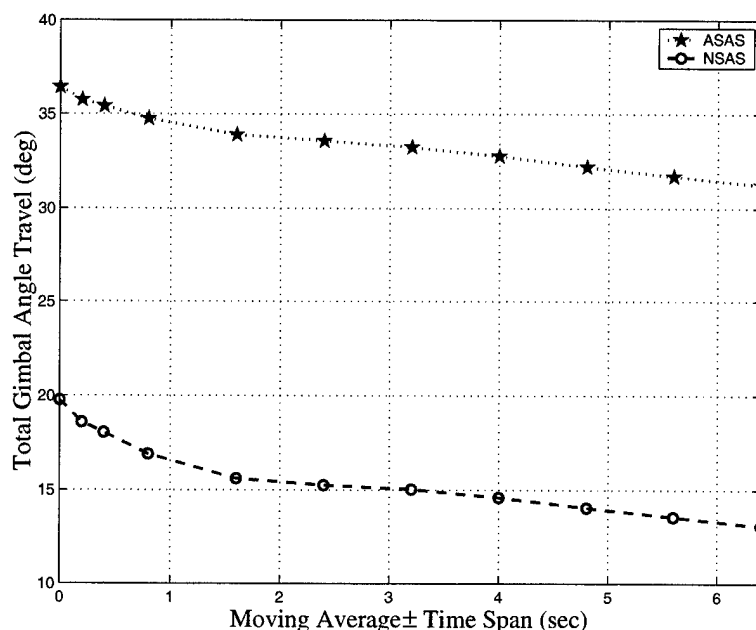
Figure 7-52: Total Gimbal Angle Travel Comparison for Varying Bias Moving Average ± Time Spans

The *NSAS* controller also shows a very strong trade relationship between gimbal activity and bending moment performance. One example of this trade is at the moving average ± time span of 4 seconds. At this point, an increase of just over 1% in bending moments produces an approximate decrease of 26% in total gimbal angle travel and 30% in gimbal angle deflection. These types of gimbal activity reductions are quite significant and could be very useful in the pursuit of lowered gimbal activity for launch vehicles operating with MPC control systems.

This chapter has examined three separate methods of trading bending moment performance for gimbal activity. These methods, controller closed-loop bandwidth manipulation, wind disturbance signal processing and bias correction signal processing, can be implemented individually or in combination with each other, depending on the vehicle restrictions present in each situation. Although each method has been explored individually, no combination studies have been conducted, which may yield new insights into the problem of high input activity for MPC-controlled systems.

[This page intentionally left blank.]

Chapter 8

Simulation Comparison Results

A wide range of comparison trial runs were completed to fully evaluate the performance of both MPC controllers and the baseline controller. As described in Chapters 1 and 5, the space launch industry currently relies upon a balloon-based system to measure the winds during launch operations. This balloon measurement process yields wind information that is an average of 90 minutes old at the time of launch. During pre-launch preparations, this wind information is checked against a set of go/no go criteria to ensure that certain wind limits are not violated. In addition, several modern load relief systems use a polynomial representation of the wind in this pre-launch process to design a trajectory which minimizes certain vehicle metrics, such as α and $Q\alpha$. The vehicle then follows this trajectory during flight, using a standard pitch controller. The earliest forms of these load relief systems used 3rd order fits to model the wind. More recent upgrades to these systems utilize 6th and 15th order fits.

8.1 Full Wind Set Comparisons

A comprehensive wind set of over 3,000 wind profiles was used for the controller comparisons. To simulate a time lapse between wind measurement and the actual launch, the wind sets were analyzed and pairs of wind profiles occurring 90 minutes apart, $\pm$ 15 minutes, were selected. 38 of these pairs were found in the ETR wind set, and 156 pairs were found in the WTR wind set. During simulation, the vehicle would

follow a commanded trajectory designed in response to the earlier wind of the pair, while the later wind would be applied to the vehicle during flight. All comparisons involving time delays were conducted using these wind pairs.

The benefits of only implementing an MPC controller or a LIDAR wind sensor individually were also investigated. To evaluate the possible benefits of using a LIDAR wind sensor alone without upgrading the on-board vehicle load-relief system, simulations were conducted against all wind profiles for each controller, using the trajectory designed to minimize α for that particular wind. It is important to recall that while the baseline controller follows this pre-planned trajectory for the full duration of the flight, the MPC controllers transition into an α following load-relief mode upon passing a Q level of 200 psf. To evaluate the possible benefits of using a MPC controller alone without improving the wind measurement equipment or process, the MPC simulations were flown using the time lapse wind pairs described previously. While the later wind profile would affect the vehicle during flight, the earlier wind profile would be loaded into the MPC controller's prediction process.

Finally, to show the benefits of an MPC controller, utilizing wind information provided by a LIDAR wind sensor, both MPC controllers were flown against all wind profiles while using current wind information for the controller's prediction process. The data resulting from these simulation runs is summarized in the following pages. The five metrics used in previous sections, peak α, peak $Q\alpha$, peak bending moments, peak gimbal angle deflection and total gimbal angle travel are shown, for the wind sets from each test range. Delay and no delay results are shown for five different controller and wind fidelity combinations, with a small graphical offset between each for added clarity. Both the average and standard deviation of each metric at each point are shown. The five combinations include the baseline controller following trajectories designed using the actual wind profile, denoted as Inf, as well as trajectories designed using 15th order and 6th order wind fits. The remaining two combinations are the $ASAS$ and $NSAS$ MPC controllers, following actual wind profiles, both present and delayed.
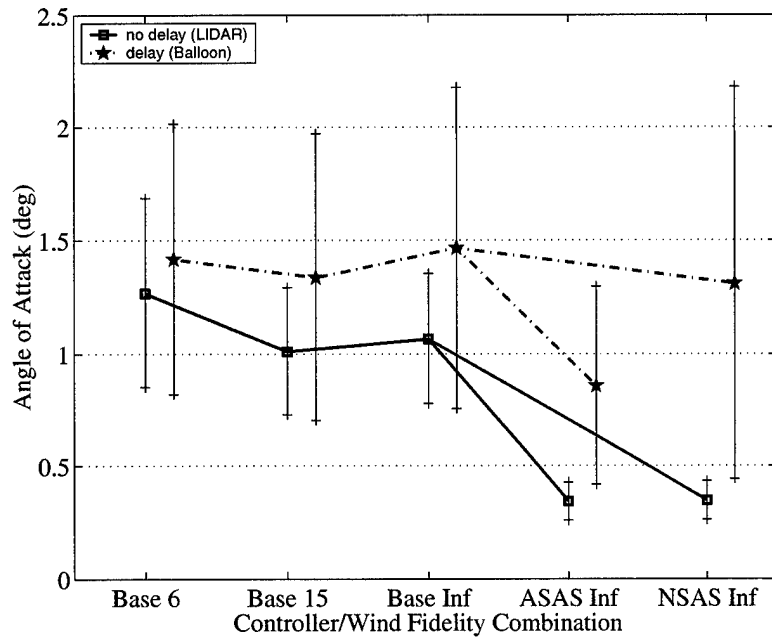
Figure 8-1: Controller Peak α Comparison for ETR
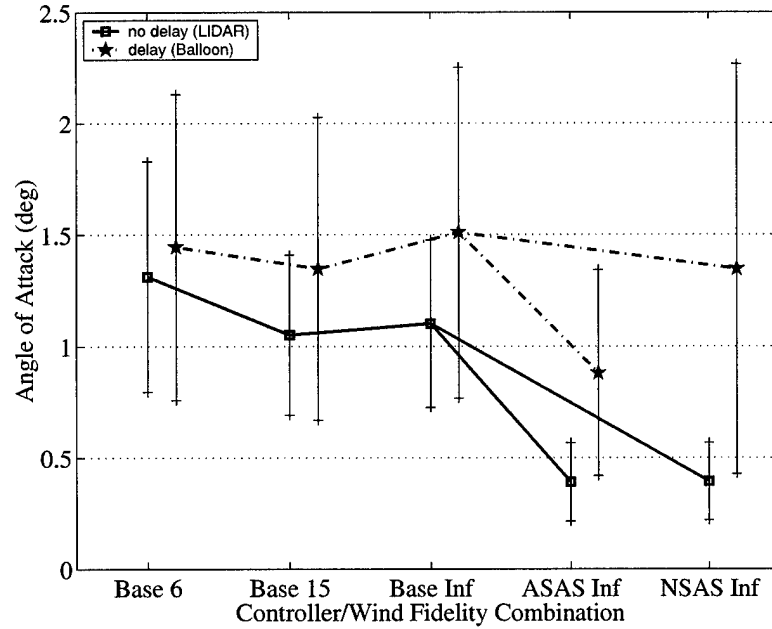
Figure 8-2: Controller Peak α Comparison for WTR
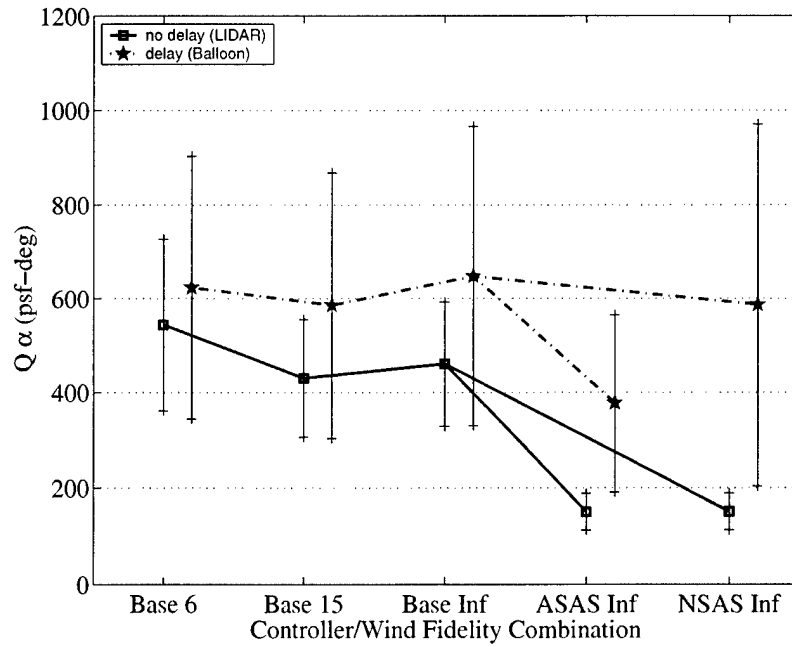
Figure 8-3: Controller Peak $Q\alpha$ Comparison for ETR
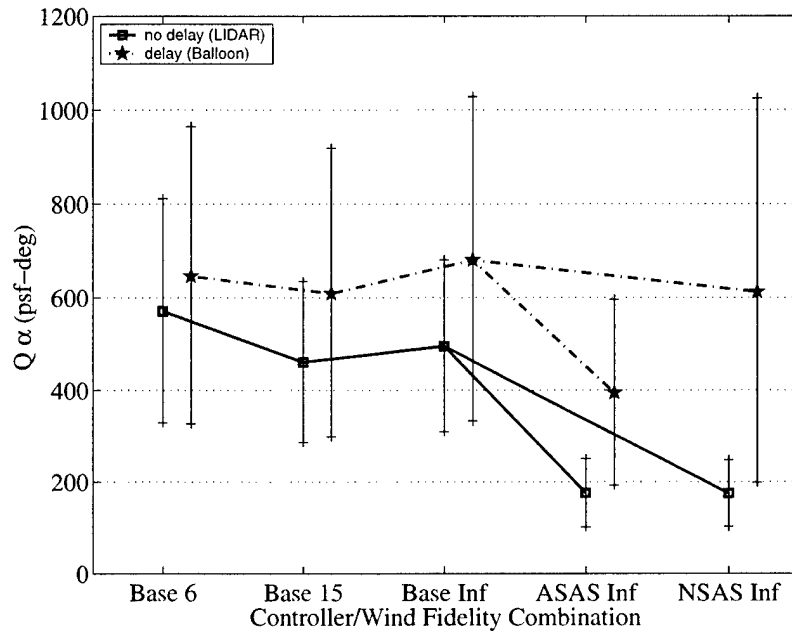
Figure 8-4: Controller Peak $Q\alpha$ Comparison for WTR

134
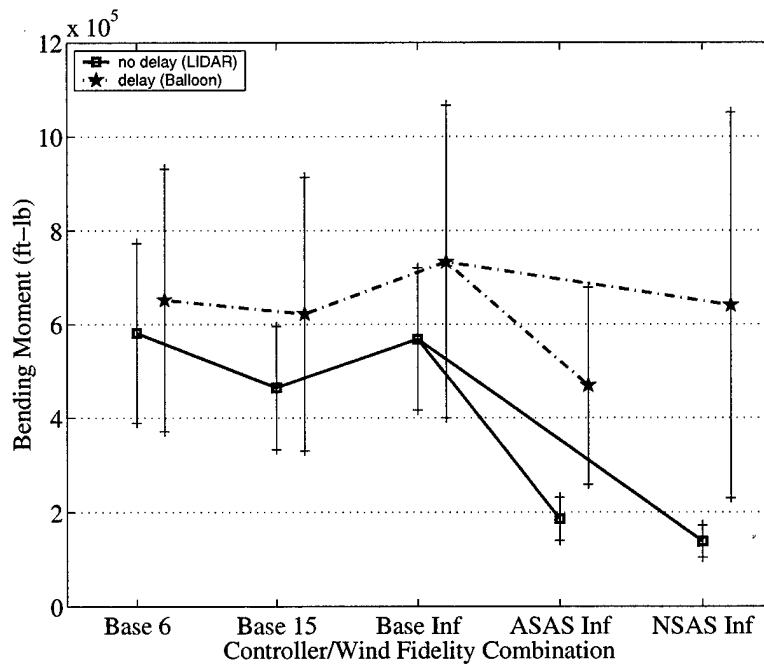
Figure 8-5: Controller Peak Bending Moment Comparison for ETR
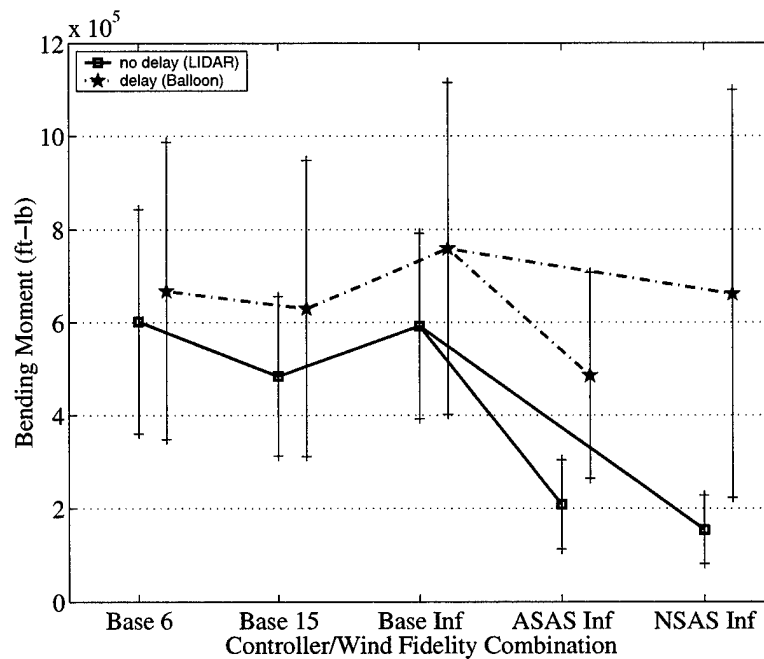
Figure 8-6: Controller Peak Bending Moment Comparison for WTR
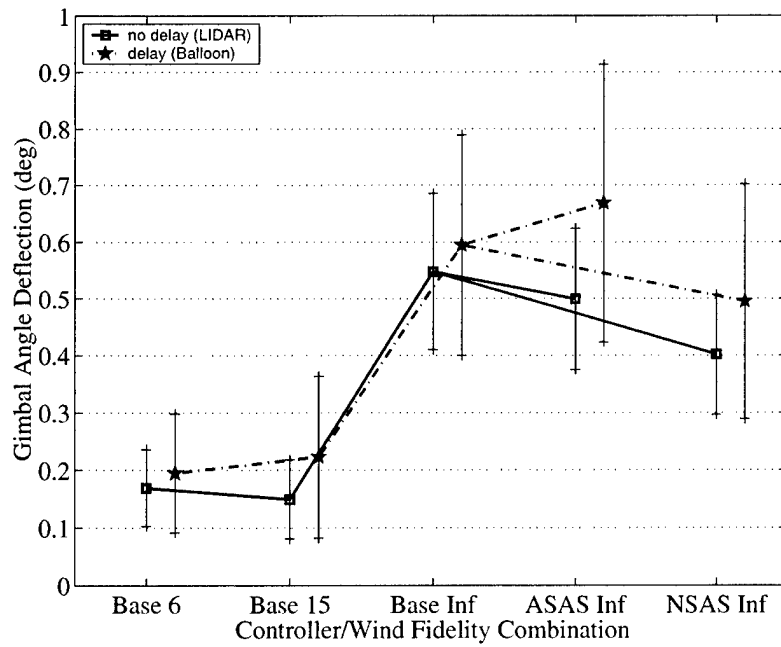
Figure 8-7: Controller Peak Gimbal Angle Comparison for ETR
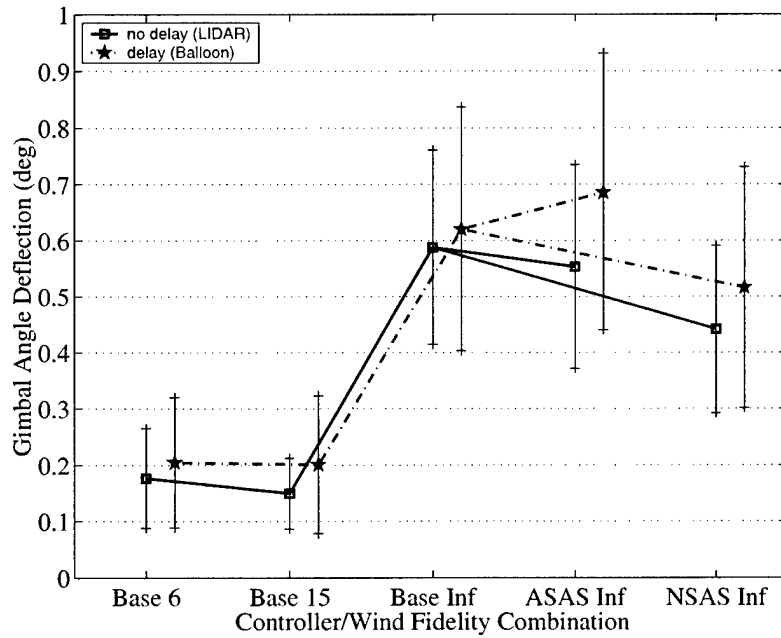
Figure 8-8: Controller Peak Gimbal Angle Comparison for WTR
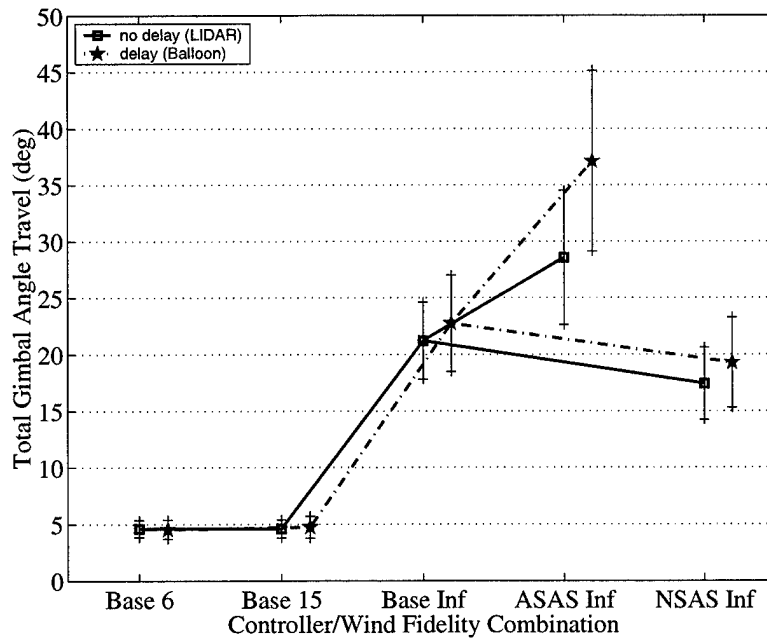
Figure 8-9: Controller Total Gimbal Angle Travel Comparison for ETR
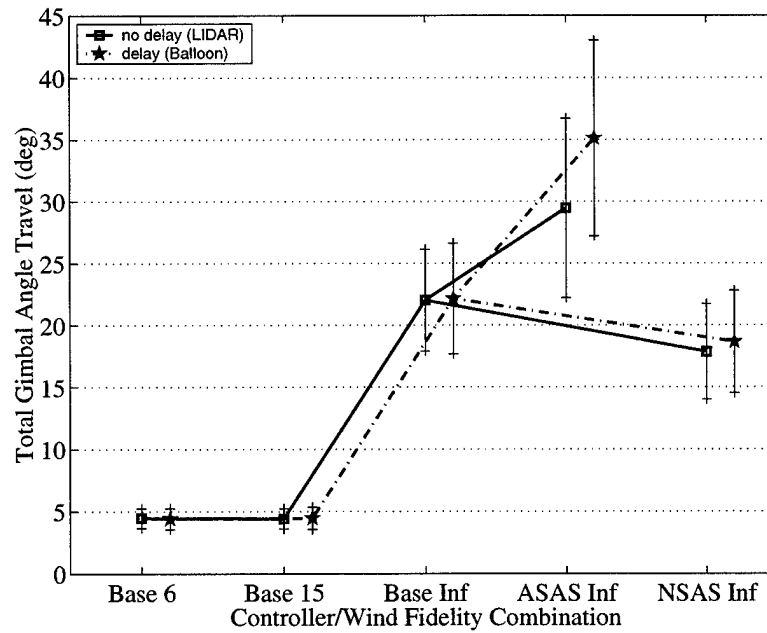
Figure 8-10: Controller Total Gimbal Angle Travel Comparison for WTR

% Reduction using $ASAS$ Controller with Actual, No Delay ETR Winds:						
	Base 6 D	Base 15 D	Base Inf D	Base 6 ND	Base 15 ND	Base Inf ND
α	75.79	74.33	76.60	72.95	66.07	67.82
$Q\alpha$	75.94	74.38	76.84	72.42	65.13	67.42
B.M.	71.45	70.10	74.62	67.99	59.95	67.25
G.A.D.	-156.59	-124.25	15.98	-195.35	-234.55	8.80
T.G.A.T.	-528.53	-503.26	-25.59	-520.57	-520.22	-34.63

% Reduction using $ASAS$ Controller with Actual, No Delay WTR Winds:						
	Base 6 D	Base 15 D	Base Inf D	Base 6 ND	Base 15 ND	Base Inf ND
α	73.01	71.05	74.18	70.27	62.87	64.59
$Q\alpha$	72.77	71.10	74.17	69.17	61.80	64.49
B.M.	68.80	66.92	72.54	65.37	56.97	64.82
G.A.D.	-170.49	-175.77	10.80	-213.00	-270.14	5.90
T.G.A.T.	-568.35	-561.82	-33.06	-561.30	-567.85	-33.79

% Reduction using $NSAS$ Controller with Actual, No Delay ETR Winds:						
	Base 6 D	Base 15 D	Base Inf D	Base 6 ND	Base 15 ND	Base Inf ND
α	75.49	74.02	76.31	72.62	65.65	67.42
$Q\alpha$	75.82	74.25	76.72	72.28	64.95	67.25
B.M.	78.72	77.71	81.08	76.14	70.15	75.59
G.A.D.	-106.37	-80.36	32.42	-137.55	-169.08	26.65
T.G.A.T.	-283.56	-268.14	23.36	-278.71	-278.49	17.84

% Reduction using $NSAS$ Controller with Actual, No Delay WTR Winds:						
	Base 6 D	Base 15 D	Base Inf D	Base 6 ND	Base 15 ND	Base Inf ND
α	72.79	70.82	73.97	70.02	62.57	64.30
$Q\alpha$	72.90	71.24	74.28	69.31	61.98	64.65
B.M.	76.83	75.44	79.61	74.29	68.05	73.88
G.A.D.	-115.94	-120.15	28.79	-149.88	-195.50	24.88
T.G.A.T.	-305.12	-301.16	19.35	-300.85	-304.82	18.90

Table 8.1: Metric % Reductions Due to MPC Controllers and LIDAR Wind Sensors for Both Test Ranges

Table 8.1 shows % reductions between each MPC controller and the baseline controller under various wind measurement and baseline controller configuration combinations. Because a percent reduction is represented by a positive number, a negative number represents a percent increase from the baseline value. The complete results from these trials can be found in Appendix C. Notice that no analysis was conducted using the MPC controllers in combination with processed wind profiles. Using smoothed wind data, as described in Chapter 7, can produce positive results in certain situations, depending on the individual launch vehicle and its gimbal movement capabilities. However, these gains cause significant increases in the α, $Q\alpha$ and bending moment metrics. For the K-1, these issues do not exist. Therefore, in order to achieve maximum reduction in bending moments, actual wind data and bias signals were used for these results. A more productive method of lowering gimbal activity without sacrificing a significant amount of bending moment performance is to smooth the bias signal, as shown at the end of Chapter 7. A set of bias smoothing results are included in the next section. Baseline controller configurations are represented by 6, 15 and Inf, which are the order of the polynomial fits used to represent the wind in each controller, respectively. Balloon and LIDAR wind measurement methods are represented by D (delay) and ND (no delay), respectively.

Many interesting observations can be made after studying these figures and tables. Significant areas of interest are the standard deviations of each controller/wind fidelity combination at different time delays, the gimbal behavior of the MPC controllers, as compared to that of various versions of the baseline controller, and the impact of the LIDAR wind sensor and the effect of wind fidelity on all controllers' performances.

The first fact to note is that in all cases, both MPC controllers are able to reduce the bending moments sustained by the launch vehicle during ascent by greater than 50%, when mated with a LIDAR wind sensor which supplies current wind data to the controller. This achievement is made even more significant when the standard deviation of the peak bending moment metric is taken into account. Both MPC controllers show bending moment standard deviations which are approximately four times smaller than equivalent delayed baseline results. This increase in performance

certainty would allow the vehicle to fly safely in rapidly changing weather conditions without increased risk, since the less conservative launch decision process would be offset by increased confidence in the vehicle performance capability.

The benefits of the LIDAR wind sensor mated with any of the simulation controllers are obvious. When the LIDAR wind sensor replaces the balloon measurement system in conjunction with a baseline controller, measurement uncertainty is reduced significantly. This is reflected in the decrease in α, $Q\alpha$ and bending moment standard deviations of approximately 50%. In general, the more advanced the control system, the more sensitive it is to wind measurement errors. This is especially evident in the case of the MPC controllers, which perform extremely well with current wind information, but much worse with delayed wind information. Relative to their nominal, no delay, performance, the MPC controllers actually perform worse than the baseline controllers with delayed wind information. In contrast, the baseline controller using a 6th order fit of the wind is affected the least by delayed information, since the low order fit acts as a filter. This filtering behavior removes all the high frequency wind information, and leaves only the general wind speed trends, which are much less likely to change significantly over a 90 minute period. As the wind profile fidelity increases, errors between the actual wind and the sensed wind used in the controller become more and more detrimental to system performance.

The gimbal behavior of each controller is also very interesting. The baseline controllers using 6th and 15th order wind fits show extremely low gimbal angle deflections and total gimbal angle travel, because of the smooth simulated wind profiles. This allows the trajectory designer to produce commanded pitch trajectories which require very little gimbal activity to follow. However, when the actual wind profile is used, the pitch profile produced by the trajectory designer is very jagged with high frequency characteristics. When the baseline pitch controller attempts to follows this commanded trajectory, it requires much more gimbal activity, as shown in Figures 8-7 through 8-10. However, because the vehicle is using a pitch controller to minimize α, the extra gimbal activity does not translate efficiently into lower α and $Q\alpha$ performance. In contrast, the MPC controllers are able to translate increased gimbal

activity into lower α, $Q\alpha$ and bending moment performance, since they operate in a direct load-relief fashion, where the variable being controlled is directly related to aerodynamic loads.

Finally, it is important to realize that by design, MPC controllers produce better results vs. a traditional control system as the wind profiles become more and more erratic and extreme. Because of the wide range of wind profiles included in this work, the average results shown do not showcase the maximum potential of MPC in the face of truly extreme wind cases. However, because the average bending moment percent reduction in all cases with current wind information is well over 50%, these percent reductions can only increase as weather conditions become less stable. This would act to increase the launch envelope of any given vehicle, addressing several of the motivating concerns described in Chapter 1.

8.2 Focused Optimized MPC/Baseline Performance Comparison

The two main purposes of this comparison set are to comprehensively showcase the ability that bias correction signal smoothing has to significantly decrease gimbal activity while only slightly increasing bending moments and to highlight the comparisons between this system and the best current baseline controller available, namely the baseline controller using a 15th order polynomial fit of delayed wind profiles.

As shown previously, the baseline controller performs best when it is provided with a smoothed version of the actual wind profile. This controller/wind processing approach is similar to current leading industry launch vehicle control concepts. The smoothed wind profile allows the gimbal activity to remain low, while still providing enough wind information to the controller to reasonably address α and $Q\alpha$ concerns. This combination of low gimbal activity and relatively good α minimization makes this the best performing baseline combination that was tested, in both the delay and no delay cases. When paired with a balloon based wind measurement system, it

can be considered equivalent to the best launch vehicle control systems in existence today. When paired with a LIDAR wind measurement system, it would hold a distinct advantage over other launch vehicle control systems currently in use.

In an effort to reduce gimbal activity as much as possible, while paying the lowest cost in bending moment performance, both the *ASAS* and *NSAS* controllers were flown against the complete ETR and WTR wind profile sets while smoothing the bias correction signal using a ± 4 second moving average. The results follow, in figure and table form. In each figure, the terms Base 15 D and Base 15 ND refer to the baseline controller using a 15th order polynomial fit of the delayed and actual wind profiles, respectively. MPC Inf and MPC Inf BC refer to the MPC controllers using the actual wind profile without and with bias correction signal smoothing. The tables contain % reductions from various baseline configurations to various MPC configurations. As in the previous set of tables, a positive number indicates a percent reduction and a negative number indicates a percent increase.
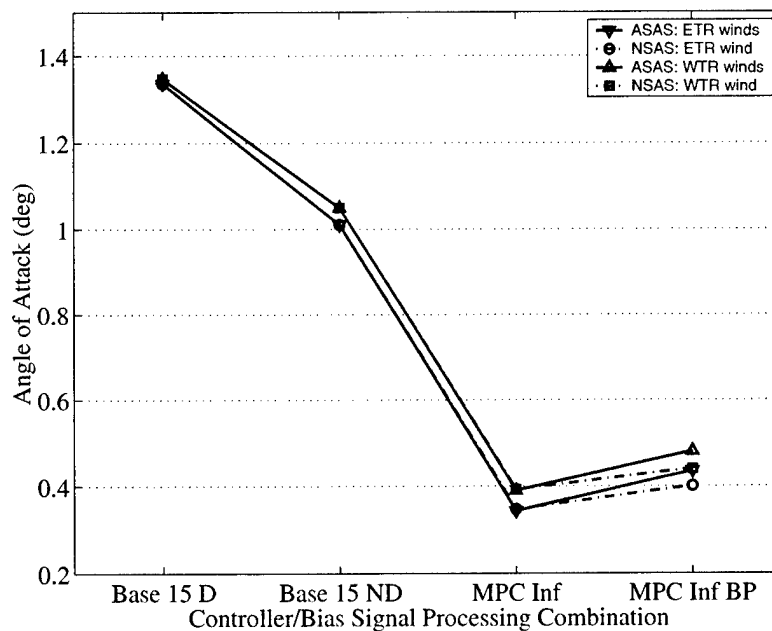
Figure 8-11: Peak α Response to Varying Controller/Bias Signal Processing Combinations
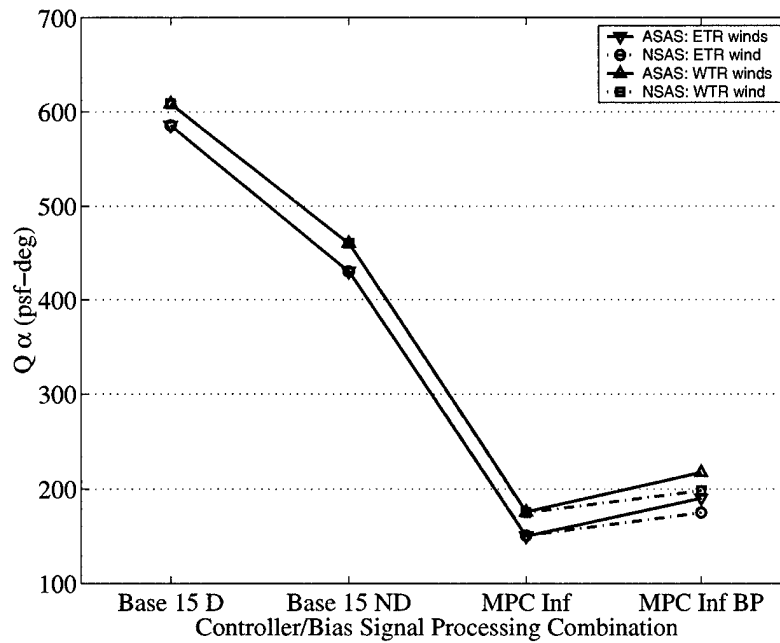
Figure 8-12: Peak $Q\alpha$ Response to Varying Controller/Bias Signal Processing Combinations
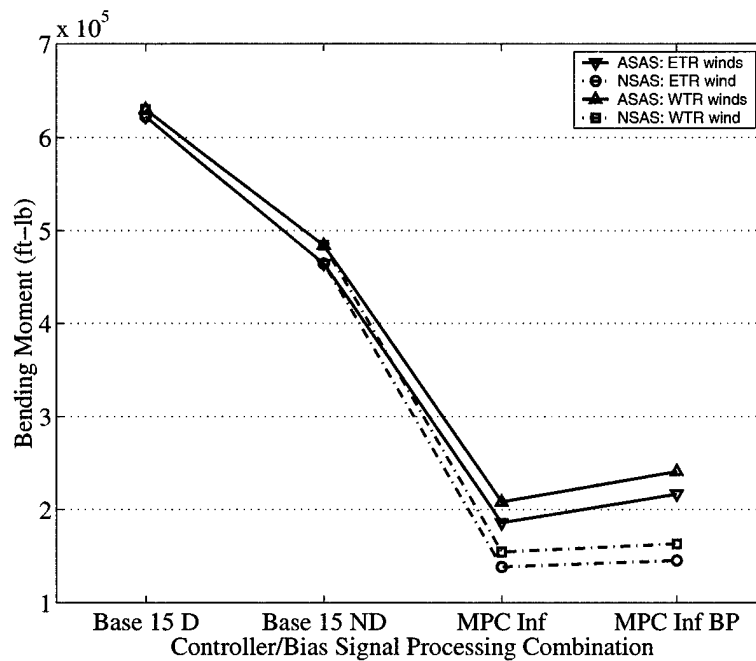
Figure 8-13: Peak Bending Moment Response to Varying Controller/Bias Signal Processing Combinations
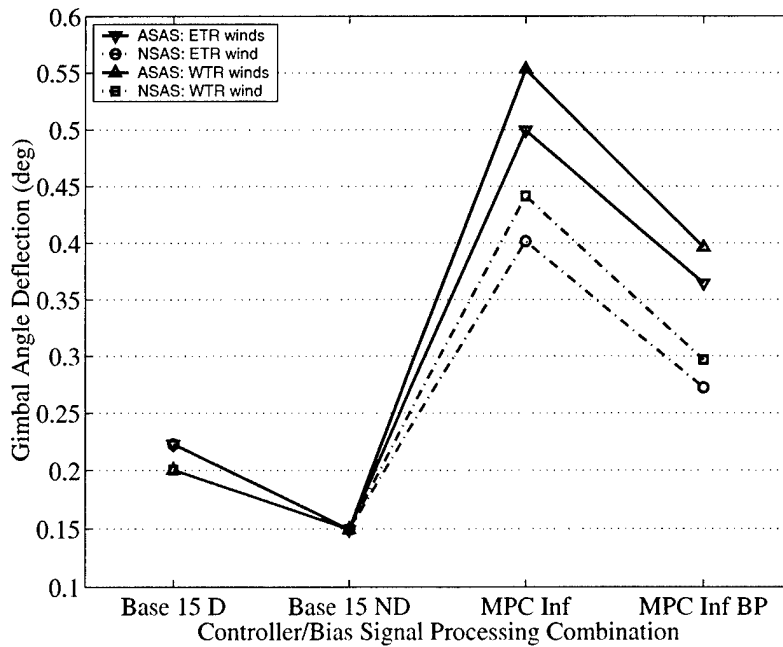
Figure 8-14: Peak Gimbal Angle Deflection Response to Varying Controller/Bias Signal Processing Combinations
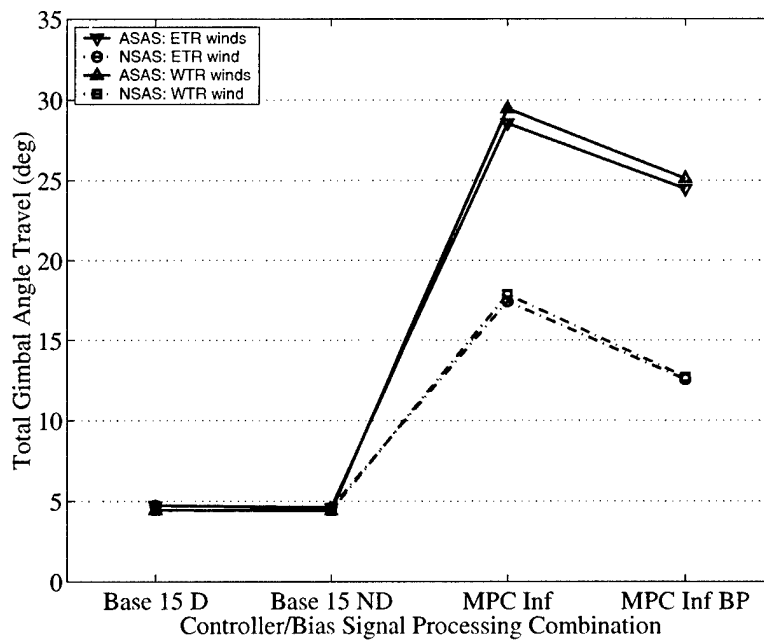
Figure 8-15: Total Gimbal Angle Travel Response to Varying Controller/Bias Signal Processing Combinations

	α	$Q\alpha$	B.M.	G.A.D.	T.G.A.T.
$ASAS \rightarrow ASAS$ BP: ETR	-26.8706	-26.8331	-16.4674	26.991	14.3278
$NSAS \rightarrow NSAS$ BP: ETR	-15.3887	-16.0626	-5.1003	32.2569	27.859
$ASAS \rightarrow ASAS$ BP: WTR	-23.2472	-23.6964	-15.6449	28.3301	14.7906
$NSAS \rightarrow NSAS$ BP: WTR	-12.0968	-13.2787	-5.7497	32.8116	28.8208

Table 8.2: MPC Metric % Reductions Due to Bias Correction Signal Smoothing

These results show that bias correction signal smoothing, or bias processing (BP) significantly decreases overall gimbal activity in both MPC controllers. The MPC controllers using smoothing are denoted with a BP. The smoothing produces an approximate 30% reduction in gimbal angle deflection for both controllers. Total gimbal angle travel is reduced by approximately 14% in the $ASAS$ controller and approximately 28% in the $NSAS$ controller. These reductions do cause a slight increase in bending moments, with larger increases in both α and $Q\alpha$ metrics as well. The $NSAS$ controller showed the smallest increase in bending moments, at approximately 5%, while the $ASAS$ showed an approximate increase of 16%. The actual results of the MPC simulations with bias signal smoothing is included in Appendix C. As stated previously, steps such as bias signal or wind processing, may or may not be necessary, depending on the individual vehicle limitations. While gimbal activity is not a factor for the K-1, it is useful to show that the tradeoff can be beneficial in certain cases. The following set of tables show that this smoothing can be implemented while still producing well over 50% reductions in α, $Q\alpha$ and bending moment metrics.

Tables 8.3 and 8.4 show the percent reductions which occur between the baseline, with and without delays, and both MPC controllers, with and without bias signal processing, in the face of winds from both test ranges. The delay and no delay results will be dealt with separately.

	α	$Q\alpha$	B.M.	G.A.D.	T.G.A.T.
Base ND → $ASAS$: ETR	66.07	65.13	59.95	-234.55	-520.22
Base ND → $NSAS$: ETR	65.65	64.95	70.15	-169.08	-278.49
Base ND → $ASAS$: WTR	62.87	61.80	56.97	-270.14	-567.85
Base ND → $NSAS$: WTR	62.57	61.98	68.05	-195.50	-304.82
Base D → $ASAS$: ETR	74.33	74.38	70.10	-124.25	-503.26
Base D → $NSAS$: ETR	74.02	74.25	77.71	-80.36	-268.14
Base D → $ASAS$: WTR	71.05	71.10	66.96	-175.77	-561.82
Base D → $NSAS$: WTR	70.82	71.24	75.44	-120.15	-301.16

Table 8.3: Metric % Reductions Due to MPC Controllers without Bias Correction Signal Processing

	α	$Q\alpha$	B.M.	G.A.D.	T.G.A.T.
Base ND → $ASAS$ BP: ETR	56.96	55.77	53.35	-144.25	-431.36
Base ND → $NSAS$ BP: ETR	60.37	59.32	68.63	-82.28	-173.05
Base ND → $ASAS$ BP: WTR	54.24	52.75	50.24	-165.28	-469.07
Base ND → $NSAS$ BP: WTR	58.04	56.93	66.21	-98.54	-188.14
Base D → $ASAS$ BP: ETR	67.44	67.51	65.17	-63.72	-416.82
Base D → $NSAS$ BP: ETR	70.02	70.12	76.58	-22.18	-165.58
Base D → $ASAS$ BP: WTR	64.32	64.26	61.74	-97.64	-463.93
Base D → $NSAS$ BP: WTR	67.28	67.42	74.02	-47.92	-185.54

Table 8.4: Metric % Reductions Due to MPC Controllers with Bias Correction Signal Processing

As previously stated, the baseline controller using delayed winds represents the best controller/wind measurement process combination currently available to the space launch industry. The implementation of the bias signal smoothing makes a very small impact on the bending moment % reduction between both MPC controllers and the baseline. The $NSAS$ change in % reductions is approximately 1% while the

ASAS change is approximately 5%. With the bias smoothing in place, the *ASAS* and *NSAS* controllers still produce approximate bending moment % reductions of 63% and 75%, respectively, over the baseline controller using delayed wind data. However, these small changes in the bending moment metrics produce significant changes in the % that the gimbal activity metrics change between the baseline and the MPC controllers. The implementation of the bias smoothing causes the *NSAS* total gimbal angle travel % increase over the baseline to change from approximately 285% to 175%, a greater than 100 percentage point decrease. Similarly, an approximate 65 percentage point decrease occurs in the *NSAS* gimbal angle deflection metric. Reductions such as this exist for the *ASAS* controller as well, although they are less prominent. This is due to the removal of direct control over the commanded gimbal angle from the MPC controller in the *ASAS* architecture.

When all simulations are provided with current wind information, the results which compare the baseline and MPC controllers highlight the advantages that MPC alone has to offer over the standard reactive control system approach. When the bias signal processing is implemented, the bending moment metric % reduction experiences an approximate 2 percentage point swing in the *NSAS* controller and an approximate 6 percentage point swing in the *ASAS* controller. When the bias smoothing is implemented, the average bending moment reduction in the *ASAS* and *NSAS* controller are approximately 52% and 67%, respectively. These small increases in bending moments yield significant decreases in gimbal activity. As a result of implementing bias smoothing, the *NSAS* total gimbal angle travel % increase changes from approximately 292% to approximately 180%, a 112 percentage point decrease. Similarly, the *NSAS* gimbal angle deflection metric % increase changes from approximately 182% to approximately 91%, a 91 percentage point decrease. These results are mirrored in the *ASAS* controller, although they are not as significant.

This research shows that MPC alone can reliably yield bending moment reductions of approximately 58% to 69%, or 52% to 67% if bias signal smoothing is implemented, depending on the controller layout. Gimbal activity is significant, but can be largely reduced if bias signal smoothing is implemented. These reductions were even larger

when the MPC controller's performance was compared to a realistic current launch vehicle control approach. Under these conditions, the MPC/LIDAR combination could reliably yield bending moment reductions of approximately 69% to 76%, or 63% to 75% if bias signal smoothing is implemented, depending on the controller layout.

8.3   Wind Measurement Noise Comparisons

A set of trials was completed to investigate the behavior of each control architecture in the face of increasing noise within the wind measurement process. 100 wind profiles from the ETR and WTR were used to create trajectory sets designed for winds containing additional wind measurement noise signals with standard deviations of 0, 2, 4, 8, 12, 16 and 20 ft/sec, respectively. Each controller was then flown following the trajectories resulting from the noisy wind profiles while only being affected by the actual wind profiles. Current wind information was used for all trials. The resulting performance metrics for each controller are shown in Figures 8-16 to 8-20.
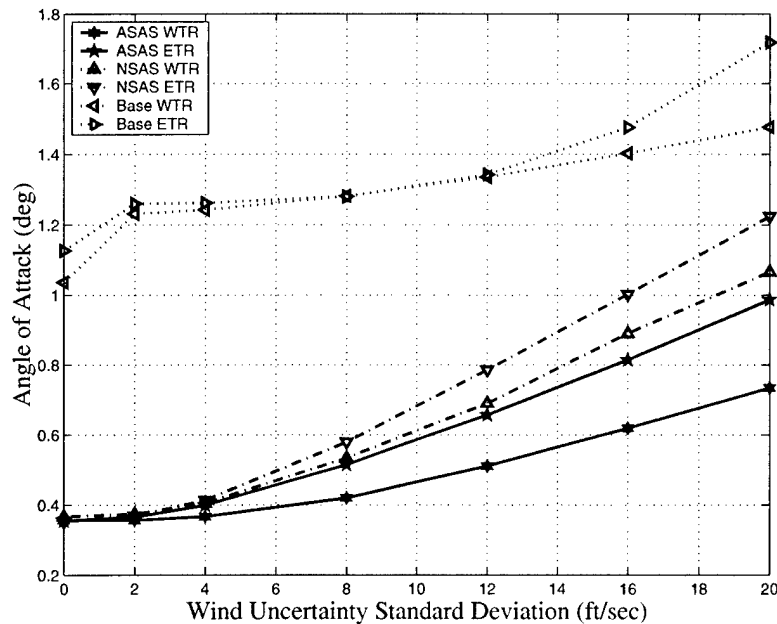
Figure 8-16: Controller Peak α Response to Wind Measurement Noise
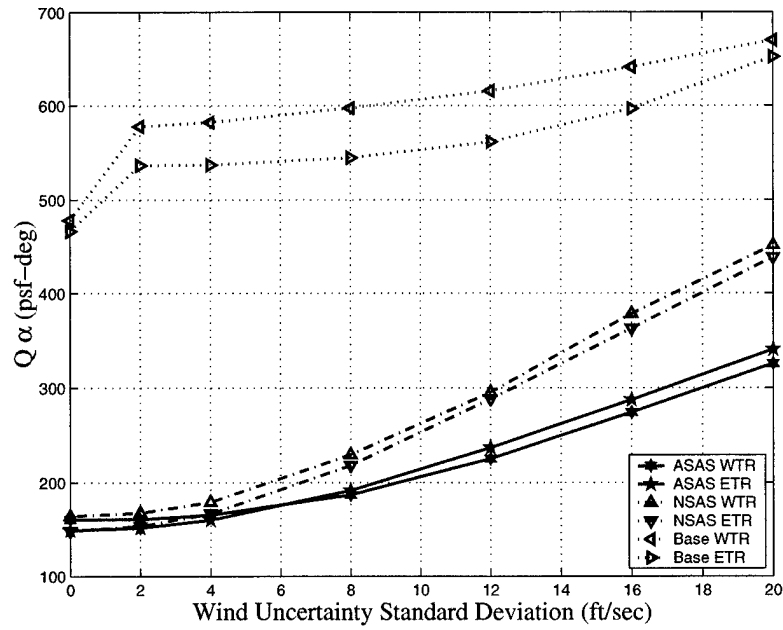
Figure 8-17: Controller Peak $Q\alpha$ Response to Wind Measurement Noise
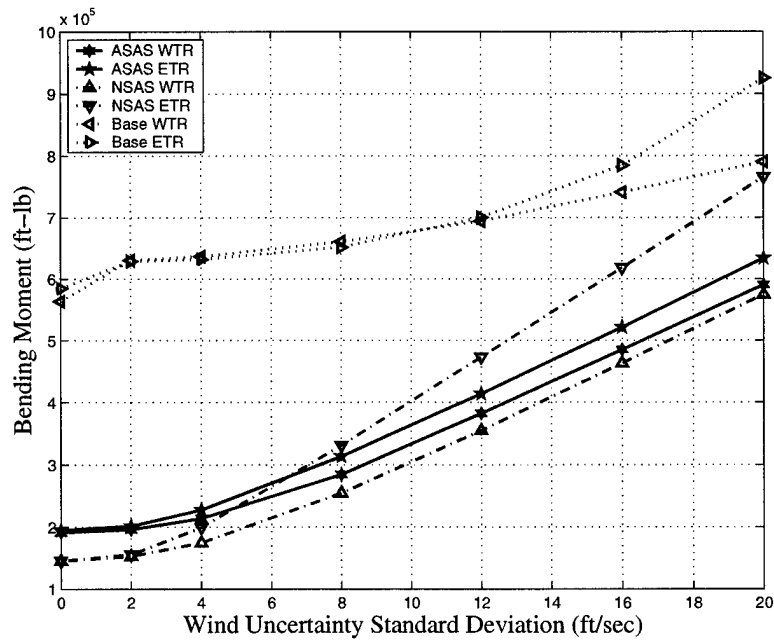
Figure 8-18: Controller Peak Bending Moment Response to Wind Measurement Noise
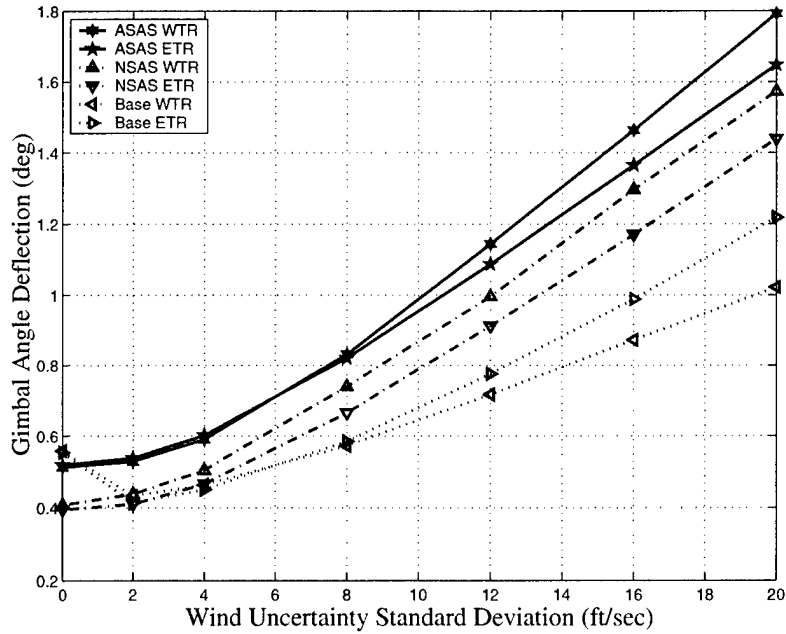
Figure 8-19: Controller Peak Gimbal Angle Response to Wind Measurement Noise
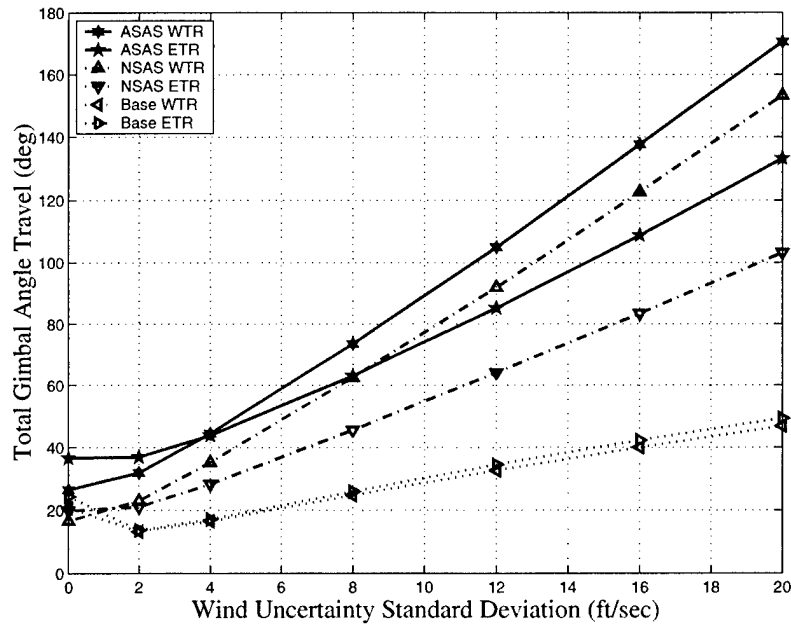
Figure 8-20: Controller Total Gimbal Angle Travel Response to Wind Measurement Noise

These figures show the general wind measurement noise sensitivity of each con-

troller type. Because the baseline controller exhibits such poor performance in no-noise conditions, it performs worse for all levels of wind measurement noise. However, relative to its no-noise performance, the baseline controller's metrics show less sensitivity to wind noise than either of the MPC controllers. Both MPC controllers exhibit α, $Q\alpha$ and bending moment metrics which are much smaller than the equivalent baseline metrics at the zero-noise point. But as noise increases, these metrics increase at a rate that is higher than the baseline rate of increase at the same points along the wind noise curve. However, both MPC controllers still show better α, $Q\alpha$ and bending moment performance than the baseline controller, even at the highest wind noise conditions tested. In fact, the MPC controllers show nearly the same or lower α, $Q\alpha$ and Bending Moment metric values at wind noise standard deviations of 20 ft/sec as the baseline controller shows at wind noise standard deviations of 0 ft/sec. These maximum wind noise levels far exceed any projected system noise conditions which might exist in a contemporary wind measurement system. Thus, the MPC controllers are shown to be superior to the baseline controller in all conceivable wind noise situations.

[This page intentionally left blank.]

Chapter 9

Conclusions and Recommendations

9.1 Conclusions

As described in Chapter 1, this thesis proposes a two-pronged approach to minimizing uncertainty and lowering launch vehicle metrics, such as α, $Q\alpha$ and bending moments. The first prong of the approach is Model Predictive Control, which leverages the accurate and timely wind disturbance data produced by the second prong of the approach, a LIDAR wind sensor. As shown in the previous chapter, while each of these can provide certain benefits by themselves, the combination of the two yields benefits far superior to the sum of their individual contributions. Without the accurate and timely wind disturbance data produced by the LIDAR wind sensor, the MPC controller cannot accurately predict the vehicle's performance, and loses much, if not all, of its advantage over the baseline controller. Therefore, these two technologies must be viewed and used as a single integrated system, replacing both the current balloon-based wind measurement system and the current launch vehicle control approach.

With the recent dramatic advances in available computational power, the technical hurdles impeding the realization of aerospace applications of MPC have become less and less significant. Current launch vehicle control systems possess more than enough computational power to implement the MPC control systems described in this thesis, assuming the matrix computations are completed before launch. The simple

matrix multiplications and additions required by this method can easily be completed during flight using today's on-board processors. While the increase in computational requirements is significant for real time calculation of the MPC matrices, near-term processors exist which will meet and surpass these needs within a matter of years.

By describing the development of two separate and unique approaches to the MPC control problem, represented by the $ASAS$ and $NSAS$ controllers, this thesis has shown that MPC is a flexible approach to launch vehicle control. The $ASAS$ controller has demonstrated that MPC can be adapted as an outer control loop around an existing stability augmentation system, while the $NSAS$ controller has showcased MPC's ability to handle both the stability augmentation and higher level guidance tasks, simultaneously.

The $ASAS$ controller shows great promise as a template for future integrations of MPC control into current launch vehicle control systems. As shown in the previous chapter, the performance improvement possibilities with this type of control system are significant. While this controller's high level of gimbal activity may be an issue for some launch vehicles, several solutions do exist for lowering this activity, such as modifications to the inner SAS or some form of α feedback signal processing.

The $NSAS$ controller reduces the MPC approach to its most simple, direct form. By avoiding the "crutch" of an inner loop SAS, MPC can be implemented in the most efficient manner possible. In cases where the MPC matrix calculations are completed before launch, this system is feasible using today's technology. Controllers which feature complete real-time MPC computation are possible using near-term computational resources. Because of its ability to control gimbal activity, maximum reductions in bending moments can be achieved using this method.

While the $ASAS$ approach appears to be the most suitable to near term applications of MPC, the $NSAS$ controller appears to hold the most promise in the long term, in the area of vehicle performance. This is not surprising, as this configuration places MPC in direct control of the input command to the plant. In future launch vehicles, with undetermined control systems and sufficient computational power, the direct application of MPC, in the form of the $NSAS$ controller, seems to be the more

154

powerful, efficient and productive implementation approach.

Implementing an MPC/LIDAR control system will certainly require additional initial investment dollars over traditional load relief and wind measurement systems. However, when compared to the benefits this system can deliver to the typical launch vehicle operator and user, these costs are highly justified and will be paid back to the operator in a relatively short time span. The increased launch capability and risk reduction inherent to the MPC approach will continue to yield benefits long after the additional investment has been recouped. Significant savings will also be realized by replacing the current manpower intensive, balloon-based process of wind measurement with the more accurate, automated and timely LIDAR-based wind measurement system.

9.2   Recommendations for Future Work

As explained earlier, both MPC controllers exhibit a bias during flight which must be corrected in order to achieve the maximum load relief performance. The source of this error is uncertain at the time of this writing, although several possible explanations exist. It is important to understand that the error may merely be a result of the unstable plant being controlled by MPC. This error does not exist when MPC is used to control a stable plant. Eliminating this bias would be useful, since it would allow the simulation to be flown without an initialization cycle, as shown in this thesis. The initialization cycle is an acceptable work-around solution to this drift problem, but certainly not the most elegant or efficient.

Another issue which should be investigated is a more realistic development of the inner SAS in the $ASAS$ controller configuration. The current simulation feeds the actual α signal, in real time, to the α SAS, which uses it to calculate the next gimbal angle command for the vehicle. This implicit assumption of accurate α knowledge may not be completely realistic. Currently, vehicle α knowledge is only available using estimation techniques of uncertain accuracy. An investigation into more accurate modelling of these methods would yield important performance data and a more

complete understanding of the performance of integrated outer-loop MPC controllers with current inner-loop stability augmentations systems.

It is also important to note that each current MPC controller parameter was selected from a subset of the complete parameter range of each variable. It is possible that a more extensive search could reveal a more advantageous parameter set. In addition, each MPC controller is also the result of a relatively crude selection process, which did not include full 6DOF, non-linear models or full ascent wind simulation. A more comprehensive investigation of these architectures using the complete simulations could yield significantly different results than were initially recorded.

The controller design process documented in this thesis operates on the concept that one controller parameter set can be found which can be used for virtually any possible wind profile. While this was demonstrated to be true through the use of over 3000 actual wind profiles occurring over nearly four decades, the subject of designing the controller for an individual wind profile was not investigated. By definition, the final controller parameter set was the average optimum result for the 127 profile design wind set. Even larger performance gains are possible if this modelling was done on an individual basis. This process could conceivably be automated and condensed into a process of only a few minutes or less, enabling maximum controller performance in the face of any given wind profile.

Appendix A

Notation Convention and Variable Declaration

A.1 Notation

Throughout this thesis, very specific notations conventions are utilized. These conventions allow the precise and succinct labelling of variables and constants.

1. Transformation matrices are referred to using the following convention:

$$T_{Sf}^{Df} \quad \text{where:} \quad Sf = Source\ frame$$
$$Df = Destination\ frame$$

   An example of this naming convention is shown below:

   T_e^i = The transformation matrix from the earth frame to the inertial frame.

   In addition, individual positions in a transformation matrix are referred to using their row and column numbers, enclosed in brackets, $T[row][column]$.

An example of this naming convention is shown below:

$T[2][3]$ = The term located in the second row and in the third column of the transformation matrix, T.

2. Vectors are referred to using the following naming convention:

$$\vec{V}_{Vs}^{Fwrt}\Big|_{Fex} \qquad \text{where:} \qquad V = Vector\ type$$

$$Vs = Vector\ source$$

$$Frwt = Frame\ with\ respect\ to$$

$$Fex = Frame\ expressed\ in$$

An example of this naming convention is shown below:

$\dot{\vec{v}}_{b}^{e}\Big|_{I}$ = The acceleration of the body, with respect to the earth frame, expressed in the inertial frame.

A.2 Variable Declaration

Throughout this thesis, many variable designations have been used. The description and units of these are included below. Note that although the standard angular units in the simulations are radians, degrees were used when analyzing the data for simplicity.

Variable Designation	Variable Description	Variable Units
Vector Type		
$\dot{\vec{v}}$	Acceleration	ft/sec^2
$\dot{\vec{\omega}}$	Angular Acceleration	rad/sec^2
$\vec{\omega}$	Angular Velocity	ft/sec

F	Force	lbf
M	Moment	ft-lb
$\vec{p}$	Position	ft
$\vec{v}$,V	Velocity	ft/sec

Vector Source

Aref	(see below)	-
a	Air	-
b	Body	-
e	Earth Frame	-
eng	Engine	-
g	Gravity	-
lg	Local Geographic Frame	-

Frames

Aref	Body Frame centered at Aero Reference Point	-
B, b	Body Frame	-
I, i, ECI	Inertial Frame	-
E, e, ECR	Earth Frame	-
LG, lg, NED	Local Geographic Frame	-

Aerodynamic Terms

α	Angle of Attack	rad
α^T	Total Angle of Attack	rad
β	Sideslip Angle	rad
C_a	Axial Force Coefficient	-
C_y	Lateral Force Coefficient	-

C_n	Normal Force Coefficient	-
C_m	Pitching Moment Coefficient	-
C_l	Rolling Moment Coefficient	-
C_n	Yawing Moment Coefficient	-
L	Reference Length	ft
p	Roll Rate	rad/sec
ϕ	Euler Roll Angle	rad
ϕ^*	Aerodynamic Roll Angle	rad
psi	Euler Yaw Angle	rad
q	Pitch Rate	rad/sec
Q, Qbar	Dynamic Pressure	lbf/ft^2 (psf)
r	Yaw Rate	rad/sec
ρ	Air Density	slug/ft^3
S	Reference Area	ft^2
SoS	Speed of Sound	ft/sec
θ	Euler Pitch Angle	rad

Round Earth Terms

GMST	Greenwich Mean Solar Time	sec
k	Earth Flattening Constant	-
Alt	Altitude	ft
Lat	Latitude	rad
Lon	Longitude	rad
re	Local Earth Radius	ft
re0	Equatorial Earth Radius	ft

Engine Terms

A_E	Nozzle Exit Area	ft^2

δ_C	Commanded Gimbal Angle	rad
δ_p	Pitch Gimbal Angle	rad
δ_y	Yaw Gimbal Angle	rad
F_T	Total Engine Thrust	lbf
Isp_{Vac}	Vacuum Specific Impulse	sec
$\dot{m}$	Mass Flow Rate	slugs/sec
TS	Throttle Setting	-
T_{Vac}	Vacuum Thrust	lbf

Mass Property Terms

I_{xx}, I_{yy}, I_{zz}	Body Moments of Inertia	slug-ft^2
I_{xy}, I_{yz}, I_{xz}	Body Cross-Products of Inertia	slug-ft^2
CG	Center of Gravity	-
CGx	CG Location in the Body X Axis Direction	ft

General

$ASAS$	MPC Controller w/ an α SAS	-
Base	Baseline PID Controller	-
ETR	Eastern Test Range, Kennedy Space Center, FL	-
KAC	Kistler Aerospace Corporation	-
LIDAR	Light Detection and Ranging	-
LQR	Linear Quadratic Regulator	-
LTI	Linear Time Invariant	-
MPC	Model Predictive Control	-
$NSAS$	MPC Controller w/o a SAS	-
PID	Proportional-plus-Integrator-plus-Derivative	-
SAS	Stability Augmentation System	-
TWD	Tail-Wags-Dog	-

| WTR | Western Test Range, Vandenburg AFB, CA | - |
| 6DOF | 6 Degree of Freedom | - |

Appendix B

Simulation S-Functions

This appendix contains a short summary of the equations executed by each S-Function. All necessary matrix transpose operations can be assumed and will not be specifically mentioned. All matrix equations will use the $T[row][column]$ convention illustrated in Appendix A. All symbols and vector conventions are also explained in Appendix A. All angles are modified to fall below 2π.

accel

$$\left.\dot{\vec{v}}^{\,i}_{\,b}\right|_I = \frac{T^i_b \vec{F}_{a+eng+g}\big|_B}{m} \tag{B.1}$$

$$\left.\dot{\vec{\omega}}^{\,i}_{\,b}\right|_B = I^{-1}\left(\vec{M}^b_{a+eng}\big|_B - \left.\vec{\omega}^{\,i}_{\,b}\right|_B \times I\left.\vec{\omega}^{\,i}_{\,b}\right|_B\right) \tag{B.2}$$

eulerRates

$$\dot{\psi} = [q\sin(\phi) + r\cos(\phi)]/\cos(\theta) \tag{B.3}$$

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi) \tag{B.4}$$

$$\dot{\phi} = p + q\tan(\theta)\sin(\phi) + r\tan(\theta)\cos(\phi) \tag{B.5}$$

envGMST

$$GMST\angle = GMST\angle(t=0) + \omega_e * \Delta time \tag{B.6}$$

This angle is used to perform a simple rotation about the Z axis to produce the current T_i^e.

xformEuler

$$
\begin{aligned}
T_i^b[1][1] &= \cos(\theta)\cos(\psi) \\
T_i^b[1][2] &= \cos(\theta)\sin(\psi) \\
T_i^b[1][3] &= -\sin(\theta) \\
T_i^b[2][1] &= \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) \\
T_i^b[2][2] &= \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) \\
T_i^b[2][3] &= \sin(\phi)\cos(\theta) \\
T_i^b[3][1] &= \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\
T_i^b[3][2] &= \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\
T_i^b[3][3] &= \cos(\phi)\cos(\theta)
\end{aligned}
\tag{B.7}
$$

$$T_e^b = T_i^b T_e^i \tag{B.8}$$

$$T^e_{lg}[1][1] = -\cos(Lon)\sin(Lat)$$

$$T^e_{lg}[1][2] = -\sin(Lon)$$

$$T^e_{lg}[1][3] = -\cos(Lon)\cos(Lat)$$

$$T^e_{lg}[2][1] = -\sin(Lon)\sin(Lat)$$

$$T^e_{lg}[2][2] = \cos(Lon) \qquad\qquad (B.9)$$

$$T^e_{lg}[2][3] = -\sin(Lon)\cos(Lat)$$

$$T^e_{lg}[3][1] = \cos(Lat)$$

$$T^e_{lg}[3][2] = 0$$

$$T^e_{lg}[3][3] = -\sin(Lat)$$

$$T^b_{lg} = T^b_e T^e_{lg} \qquad\qquad (B.10)$$

velocityOmega

$$\left.\vec{\omega}^i_b\right|_I = T^i_b \left.\vec{\omega}^i_b\right|_B \qquad\qquad (B.11)$$

$$\left.\vec{\omega}^i_e\right|_I = [\,0\ \ 0\ \ \omega_e\,] \qquad\qquad (B.12)$$

$$\left.\vec{\omega}^i_e\right|_B = T^b_i \left.\vec{\omega}^i_e\right|_I \qquad\qquad (B.13)$$

$$\left.\vec{\omega}^e_b\right|_B = \left.\vec{\omega}^i_b\right|_B - \left.\vec{\omega}^i_e\right|_B \qquad\qquad (B.14)$$

$$\left.\vec{v}^i_b\right|_E = \left.\vec{v}^i_b\right|_I + \left.\vec{\omega}^i_e\right|_I \times \left.\vec{p}^i_b\right|_I \qquad\qquad (B.15)$$

$$r_p = \sqrt{\left.\overrightarrow{p}_b^i\right|_I(x)^2 + \left.\overrightarrow{p}_b^i\right|_I(y)^2}$$ (B.16)

$$\dot{Lat} = k \frac{\left(r_p * \left.\overrightarrow{v}_b^i\right|_E(z) - \frac{\left.\overrightarrow{p}_b^i\right|_I(z)\left(\left.\overrightarrow{p}_b^i\right|_I(x)\left.\overrightarrow{v}_b^i\right|_E(x) + \left.\overrightarrow{p}_b^i\right|_I(y)\left.\overrightarrow{v}_b^i\right|_E(y)\right)}{r_p}\right)}{k^2 r_p^2 + \left.\overrightarrow{p}_b^i\right|_I(z)^2}$$ (B.17)

$$\dot{Lon} = \frac{\left.\overrightarrow{p}_b^i\right|_I(x) * \left.\overrightarrow{v}_b^i\right|_E(y) - \left.\overrightarrow{p}_b^i\right|_I(y) * \left.\overrightarrow{v}_b^i\right|_E(x)}{r_p^2}$$ (B.18)

$$
\begin{aligned}
\left.\overrightarrow{\omega}_{lg}^e\right|_E[1] &= \dot{Lat} * \sin(Lon) \\
\left.\overrightarrow{\omega}_{lg}^e\right|_E[2] &= -\dot{Lat} * \cos(Lon) \\
\left.\overrightarrow{\omega}_{lg}^e\right|_E[3] &= \dot{Lon}
\end{aligned}
$$ (B.19)

$$\left.\overrightarrow{\omega}_{lg}^e\right|_B = T_e^b \left.\overrightarrow{\omega}_{lg}^e\right|_E$$ (B.20)

$$\left.\overrightarrow{\omega}_b^{lg}\right|_B = \left.\overrightarrow{\omega}_b^e\right|_B - \left.\overrightarrow{\omega}_{lg}^e\right|_B$$ (B.21)

$$\left.\overrightarrow{v}_b^i\right|_B = T_i^b \left.\overrightarrow{v}_b^i\right|_I$$ (B.22)

$$\left.\overrightarrow{v}_e^i\right|_I = \left.\overrightarrow{\omega}_e^i\right|_I \left.\overrightarrow{p}_b^i\right|_I$$ (B.23)

$$\left.\overrightarrow{v}_e^i\right|_B = T_i^b \left.\overrightarrow{v}_e^i\right|_I$$ (B.24)

$$\left.\overrightarrow{v}_b^e\right|_B = \left.\overrightarrow{v}_b^i\right|_B - \left.\overrightarrow{v}_e^i\right|_I$$ (B.25)

$$\vec{v}_b^e\big|_{LG} = T_b^{lg}\vec{v}_b^e\big|_B \tag{B.26}$$

$$\vec{v}_b^e\big|_E = T_b^e\vec{v}_b^e\big|_B \tag{B.27}$$

$$\vec{v}_a^e\big|_B = T_{lg}^b\vec{v}_a^e\big|_{LG} \tag{B.28}$$

$$\vec{v}_b^a\big|_B = \vec{v}_b^e\big|_B - \vec{v}_a^e\big|_B \tag{B.29}$$

gravityShepperd

This model produces $\dot{\vec{v}}_g\big|_E$ following the procedures outlined in [8] and [9].

$$\dot{\vec{v}}_g\big|_B = T_e^b\dot{\vec{v}}_g\big|_E \tag{B.30}$$

$$weight = m|\dot{\vec{v}}_g| \tag{B.31}$$

$$\vec{F}_g\big|_B = m\dot{\vec{v}}_g\big|_B \tag{B.32}$$

aeroParameters

$$V = |\vec{v}_b^a\big|_B| \tag{B.33}$$

$$\beta = \sin^{-1}(\vec{v}_b^a\big|_B(2)/V) \tag{B.34}$$

$$\alpha^T = \cos^{-1}(\vec{v}_b^a\big|_B(1)/V) \tag{B.35}$$

167
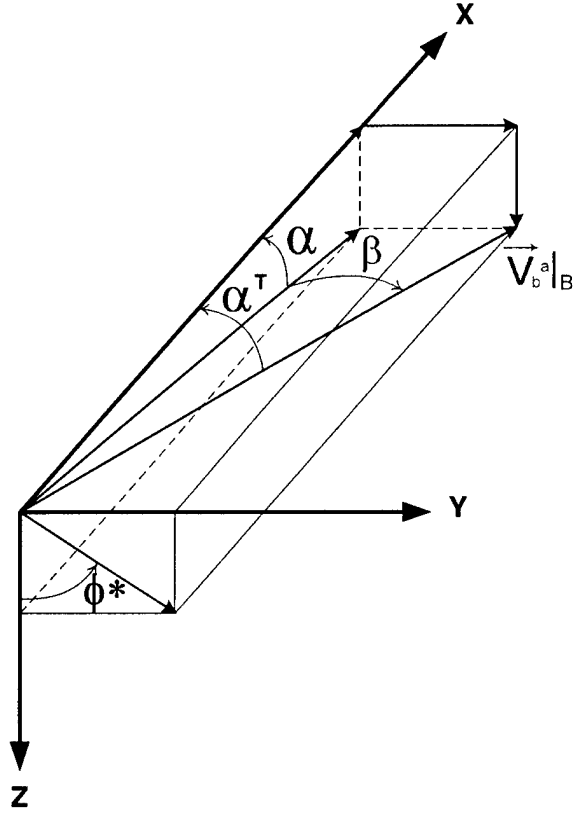
Figure B-1: Aerodynamic Angles

$$\alpha = \tan^{-1}(\overrightarrow{v}^{\,a}_{b}\big|_{B}(3)/\overrightarrow{v}^{\,a}_{b}\big|_{B}(1)) \tag{B.36}$$

$$\phi^* = \tan^{-1}(\overrightarrow{v}^{\,a}_{b}\big|_{B}(2)/\overrightarrow{v}^{\,a}_{b}\big|_{B}(3)) \tag{B.37}$$

$$mach = V/SoS \tag{B.38}$$

$$Q = 1/2\rho\, V^2 \tag{B.39}$$

atmosphereUS76

This is a 1976 U.S. standard atmospheric model, as detailed in [6].

xyz2lat_lon_alt

$$re = \sqrt{\left(\frac{re0}{1 + \left(\frac{\left.\vec{p}_b^i\right|_E(z)}{\left.\vec{p}_b^i\right|_E \bullet \left.\vec{p}_b^i\right|_E}\right)^2}\right)^2 \left(\frac{1}{k-1}\right)} \tag{B.40}$$

$$Alt = \left|\left.\vec{p}_b^i\right|_E\right| - re \tag{B.41}$$

$$Lon = \arctan\left(\frac{\left.\vec{p}_b^i\right|_E(y)}{\left.\vec{p}_b^i\right|_E(x)}\right) \tag{B.42}$$

$$Lat = \arctan\left(\frac{\left.\vec{p}_b^i\right|_E(z)}{\left(\sqrt{\left.\vec{p}_b^i\right|_E(x)^2 + \left.\vec{p}_b^i\right|_E(y)^2}\right) k \cos(Lat)}\right) Alt(1-k) \tag{B.43}$$

engine

$$\dot{m} = \frac{TS * T_{Vac}}{Isp_{Vac} * |\dot{\vec{v}}_g|} \tag{B.44}$$

$$F_T = TS * T_{Vac} - AirPressure * A_E \tag{B.45}$$

$$\begin{aligned}
\left.\vec{F}_{eng}\right|_B[x] &= \cos(\delta_p)\cos(\delta_y)F_T \\
\left.\vec{F}_{eng}\right|_B[y] &= -\cos(\delta_p)\sin(\delta_y)F_T \\
\left.\vec{F}_{eng}\right|_B[z] &= \sin(\delta_p)\cos(\delta_y)F_T
\end{aligned} \tag{B.46}$$

$$\left.\vec{M}_{eng}\right|_B = \left.\vec{p}_{eng}^b\right|_B \times \left.\vec{F}_{eng}\right|_B \tag{B.47}$$

aeroK1Boost

This function uses linear interpolation to select normal, axial and lateral force co-efficients, and roll, pitch and yaw moment coefficients using mach number and α^T. These coefficients are denoted as Cn, Ca, Cy, Cl, Cm and Cn respectively.

$$F_n = QSCn \tag{B.48}$$

$$F_a = QSCa \tag{B.49}$$

$$F_l = QSCy \tag{B.50}$$

$$M_r = QSLCl \tag{B.51}$$

$$M_p = QSLCm \tag{B.52}$$

$$M_y = QSLCn \tag{B.53}$$

$$
\begin{aligned}
\left.\overrightarrow{F}_a\right|_B[x] &= F_a \\
\left.\overrightarrow{F}_a\right|_B[y] &= -F_l\cos(\phi^*) + F_n\sin(\phi^*) \\
\left.\overrightarrow{F}_a\right|_B[z] &= F_l\sin(\phi^*) + F_n\cos(\phi^*)
\end{aligned}
\tag{B.54}
$$

$$
\begin{aligned}
\left.\overrightarrow{M}_a^{Aref}\right|_B[r] &= M_r \\
\left.\overrightarrow{M}_a^{Aref}\right|_B[p] &= M_p\cos(\phi^*) + M_y\sin(\phi^*) \\
\left.\overrightarrow{M}_a^{Aref}\right|_B[y] &= -M_p\sin(\phi^*) + M_y\cos(\phi^*)
\end{aligned}
\tag{B.55}
$$

$$\left.\overrightarrow{M}_a^b\right|_B = \left.\overrightarrow{M}_a^{Aref}\right|_B + \left.\overrightarrow{p}_{Aref}^b\right|_B \times \left.\overrightarrow{F}_a\right|_B \tag{B.56}$$

Appendix C

Simulation Results

The results shown in this appendix are from the complete wind profile set, 1,141 wind profiles from the WTR and 1,927 wind profiles from the ETR.

C.1 Full Wind Set Comparison Results

Results from ETR Winds without Delay

	Base 6	Base 15	Base Inf	$ASAS$ Inf	$NSAS$ Inf
α Avg.	1.268	1.0109	1.0657	0.34298	0.34721
α St.Dev.	0.41653	0.28236	0.28821	0.083768	0.085492
$Q\alpha$ Avg.	544.0496	430.2306	460.4586	150.0318	150.7934
$Q\alpha$ St.Dev.	182.8657	124.8899	132.2233	37.7273	38.0168
B.M. Avg.	581133.8751	464495.7011	568080.7079	186039.58	138646.6724
B.M. St.Dev.	191880.8066	131364.4525	152070.0949	45653.7213	33862.486
G.A.D. Avg.	0.16908	0.14927	0.5476	0.49939	0.40166
G.A.D. St.Dev.	0.066439	0.068349	0.13759	0.12445	0.10552
T.G.A.T. Avg.	4.6018	4.6044	21.2122	28.5573	17.4272
T.G.A.T. St.Dev.	0.75429	0.79842	3.3997	5.9521	3.1971

Results from WTR Winds without Delay

	Base 6	Base 15	Base Inf	$ASAS$ Inf	$NSAS$ Inf
α Avg.	1.3115	1.0503	1.1014	0.38996	0.39315
α St.Dev.	0.51623	0.35849	0.3748	0.17667	0.17374
$Q\alpha$ Avg.	570.4877	460.4039	495.2585	175.8712	175.0687
$Q\alpha$ St.Dev.	241.2194	173.9208	185.9545	74.6883	73.0384
B.M. Avg.	601636.9623	484189.9257	592250.4283	208352.6563	154689.7569
B.M. St.Dev.	240990.6595	171837.667	199489.1667	95371.7645	73042.8821
G.A.D. Avg.	0.1767	0.14942	0.58775	0.55308	0.44154
G.A.D. St.Dev.	0.088578	0.062783	0.17337	0.18183	0.14927
T.G.A.T. Avg.	4.4559	4.4122	22.0244	29.4669	17.8614
T.G.A.T. St.Dev.	0.78769	0.80865	4.1171	7.25	3.8455

Results from ETR Winds with Delay

	Base 6	Base 15	Base Inf	$ASAS$ Inf	$NSAS$ Inf
α Avg.	1.4168	1.3363	1.4656	0.85738	1.3106
α St.Dev.	0.59751	0.63388	0.71057	0.43991	0.86828
$Q\alpha$ Avg.	623.5383	585.7086	647.8754	377.8147	586.8081
$Q\alpha$ St.Dev.	279.5638	282.389	317.9526	186.7863	383.8624
B.M. Avg.	651627.9908	622113.4911	732958.6135	468970.4965	640633.4827
B.M. St.Dev.	280245.4606	291825.462	333230.6556	209664.483	410049.2085
G.A.D. Avg.	0.19463	0.2227	0.5944	0.66837	0.49532
G.A.D. St.Dev.	0.10313	0.14074	0.19454	0.24516	0.20646
T.G.A.T. Avg.	4.5435	4.7339	22.7386	37.1063	19.289
T.G.A.T. St.Dev.	0.8316	0.97989	4.2621	8.0159	3.9941

Results from WTR Winds with Delay

	Base 6	Base 15	Base Inf	$ASAS$ Inf	$NSAS$ Inf
α Avg.	1.4447	1.3471	1.5102	0.88017	1.3462
α St.Dev.	0.68639	0.67906	0.74221	0.46195	0.91945
$Q\alpha$ Avg.	645.9825	608.6365	680.7474	394.4207	612.053
$Q\alpha$ St.Dev.	318.9512	309.9272	347.4357	201.6737	412.9823
B.M. Avg.	667723.2229	629766.3976	758800.298	485601.4973	661437.2946
B.M. St.Dev.	319384.6031	318289.0028	356519.2014	221628.7625	438570.0142
G.A.D. Avg.	0.20447	0.20056	0.62002	0.6856	0.51599
G.A.D. St.Dev.	0.11554	0.12238	0.21711	0.24581	0.21478
T.G.A.T. Avg.	4.4089	4.4524	22.1463	35.1124	18.6623
T.G.A.T. St.Dev.	0.84254	0.89813	4.4773	7.9057	4.1409

C.2 Focused Optimized MPC/Baseline Performance Comparison Results

Results of MPC Controllers with Bias Signal Processing

	$ASAS$:ETR	$NSAS$:ETR	$ASAS$:WTR	$NSAS$:WTR
α Avg.	0.43514	0.40064	0.48062	0.44071
α St.Dev.	0.10277	0.096555	0.18169	0.17674
$Q\alpha$ Avg.	190.2901	175.0148	217.5464	198.3155
$Q\alpha$ St.Dev.	47.0706	43.8483	80.0166	76.3696
B.M. Avg.	216675.4129	145718.1052	240949.144	163583.9486
B.M. St.Dev.	53240.2255	36175.5857	98192.2195	74291.8856
G.A.D. Avg.	0.3646	0.2721	0.39639	0.29667
G.A.D. St.Dev.	0.088407	0.07436	0.14833	0.10215
T.G.A.T. Avg.	24.4657	12.5722	25.1086	12.7136
T.G.A.T. St.Dev.	5.6506	2.3763	6.7018	2.799

Appendix D

Simulation Parameters

The variables contained in the simulation initialization file are shown in this appendix. The variables used to initialize the mass properties block are listed in Chapter 2.

α SAS Controller Gains

Variable Name	Variable Value	Variable Units
Ka	1.4	–
Kd	1.2	sec
Kp	0.7	–
Ks	0.1	–

Initial State Values

Variable Name	Variable Value	Variable Units
Initial Position	[0 0 0]	ft
Initial Velocity	[0 0 0]	ft/sec
Initial Angular Position	[0 0 0]	rad
Initial Angular Velocity	[0 0 0]	rad/sec
Initial Mass	26243	slugs

Engine Parameters

Variable Name	Variable Value	Variable Units
Engine Position	[34.2667 0 0]	ft
Vacuum Thrust	378000	lbf
Vacuum Isp	331.3	sec
Exit Area	18.479167	ft^2

Aerodynamic Parameters

Variable Name	Variable Value	Variable Units
Aero Reference Point	[83.333 0 0]	ft
Reference Area	201.0	ft^2
Reference Length	16.0	ft

Actuator Dynamics and Atmospheric Variables

Variable Name	Variable Value	Variable Units
ω_n	25.132	rad/sec
ζ	0.707	–
Earth Radius	20925741	ft
Gravity Inverse	0.03108	sec^2/ft

Appendix E

SIMULINK Non-Linear 6DOF Implementation

E.1 SIMULINK Model Structure

The SIMULINK operating environment allows the user to create complex simulations with any number of subsystems nested in multiple levels. This enables the implementation of highly complicated system simulations while still maintaining a simple, easy to understand interface. The following sections contain pictures of the significant systems and subsystems contained within the *NSAS*, *ASAS* and Baseline models. Because the simulations are identical except for the pitch controller in each model, only one set of of pictures will be included for the remainder of the Rocket subsystems, which are located in the *NSAS* Model section.

The Scopes subsystem exists in all three simulations, and contains SIMULINK variable plotting blocks which can be used to observe different signals during or after simulation. The three outputs from the Scopes subsystem are the bending moments, calculated at stations located at various points on the vehicle. The station used for all bending moment readings in this thesis was the furthest aft station, Station 1000, located approximately at the CG.
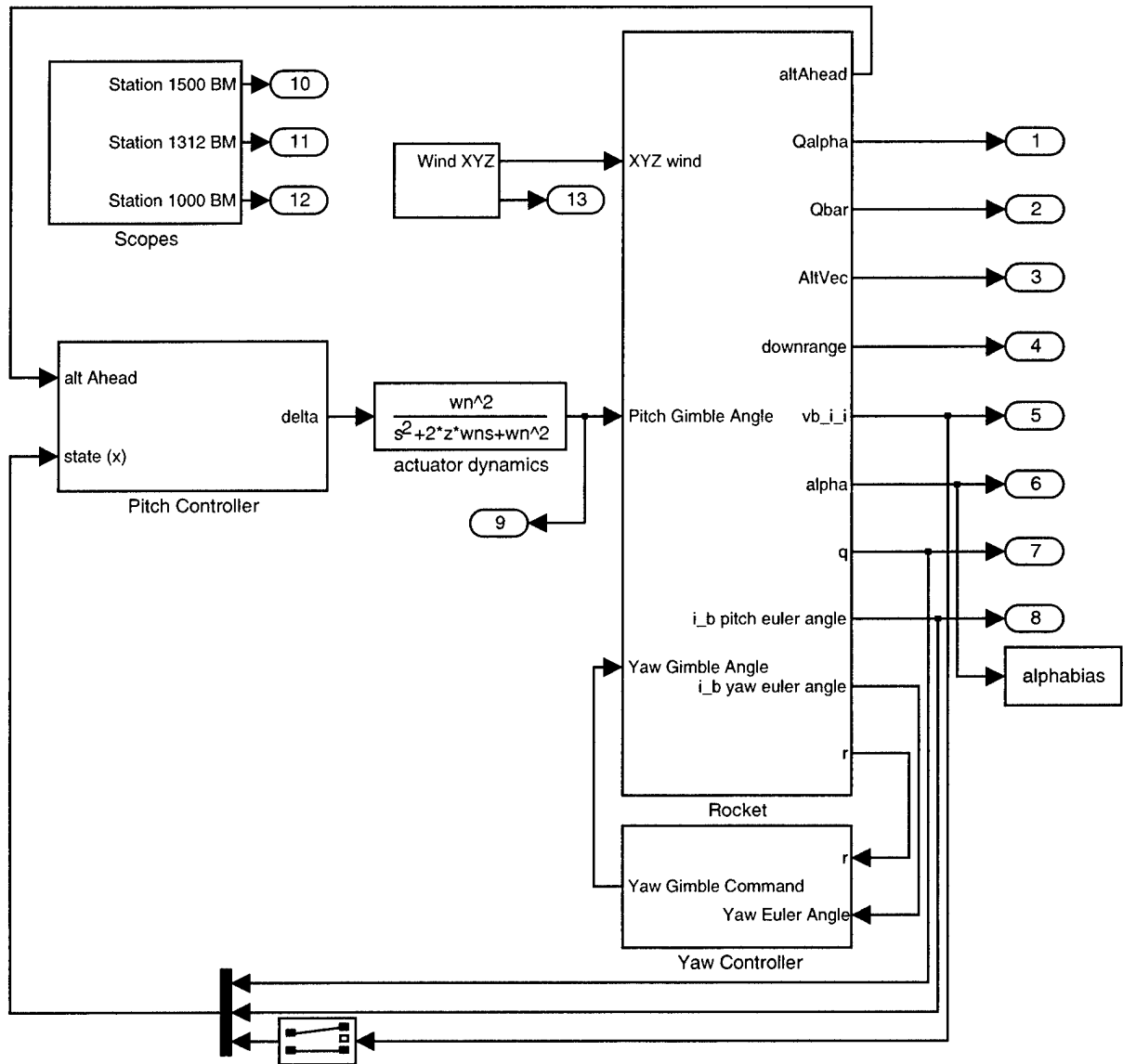
E.2 *NSAS* Model

Figure E-1: *NSAS* Model: Top Level

The Wind XYZ subsystem contains the actual wind disturbance profile which affects the vehicle during flight and is common to all three simulations. The Actuator Dynamics block is a simple, second order transfer function using the previously defined values of ω_n and ζ.
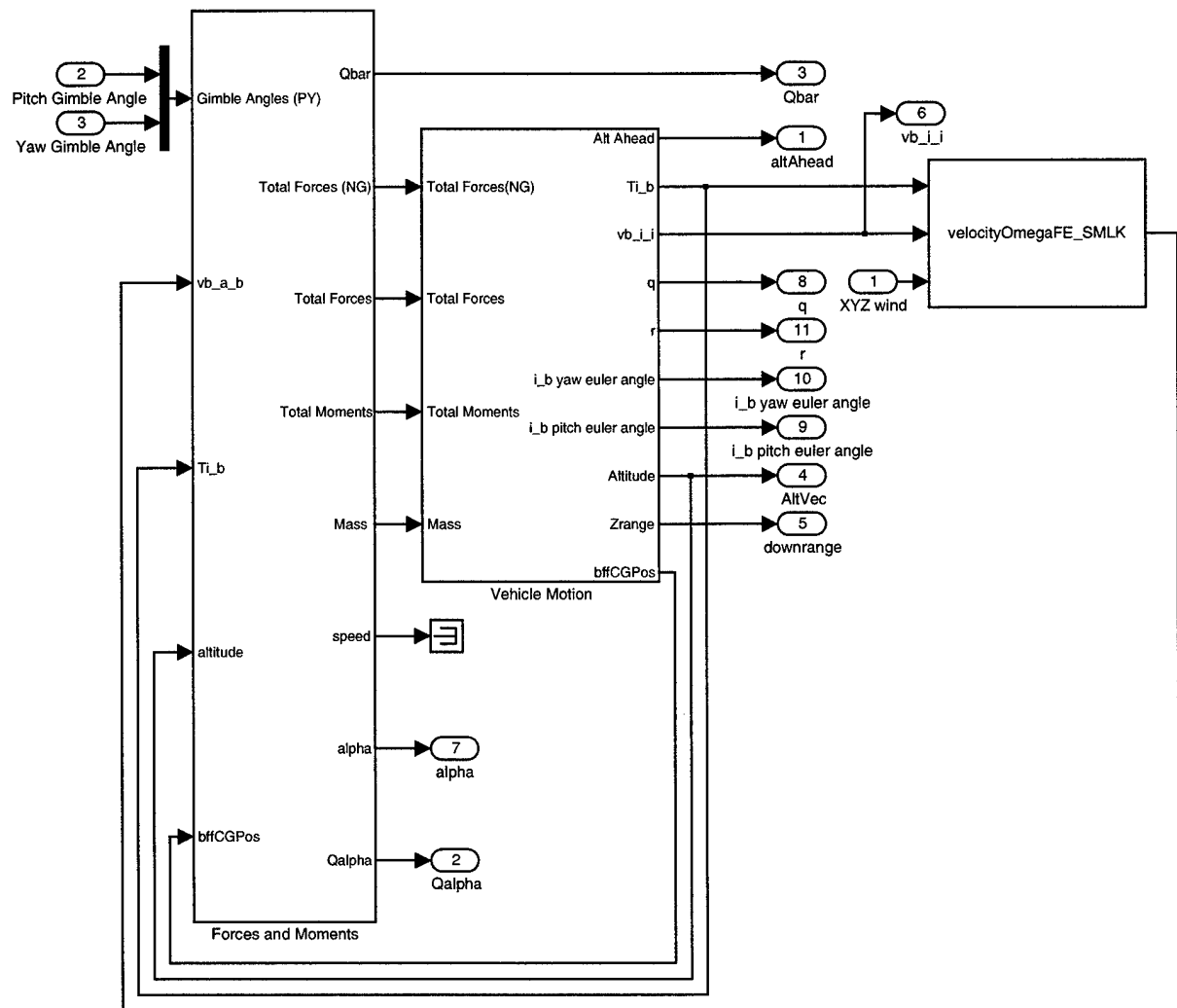
Figure E-2: *NSAS* Model: Rocket

The Rocket subsystem contains two subsystems, Forces and Moments and Vehicle Motion, as well as a flat-earth version of the S-function, velocityOmega_SMLK. bf-fCGPos is the position of the CG in the body frame. altAhead is a vector of projected vehicle altitudes. The vector contains p steps, with a step size equal to the prediction step size. Outputs are shown as ovals. If these exist in the top level of a model, the signals are passed out to the Matlab workspace. Otherwise, they pass signals up to the next subsystem level.
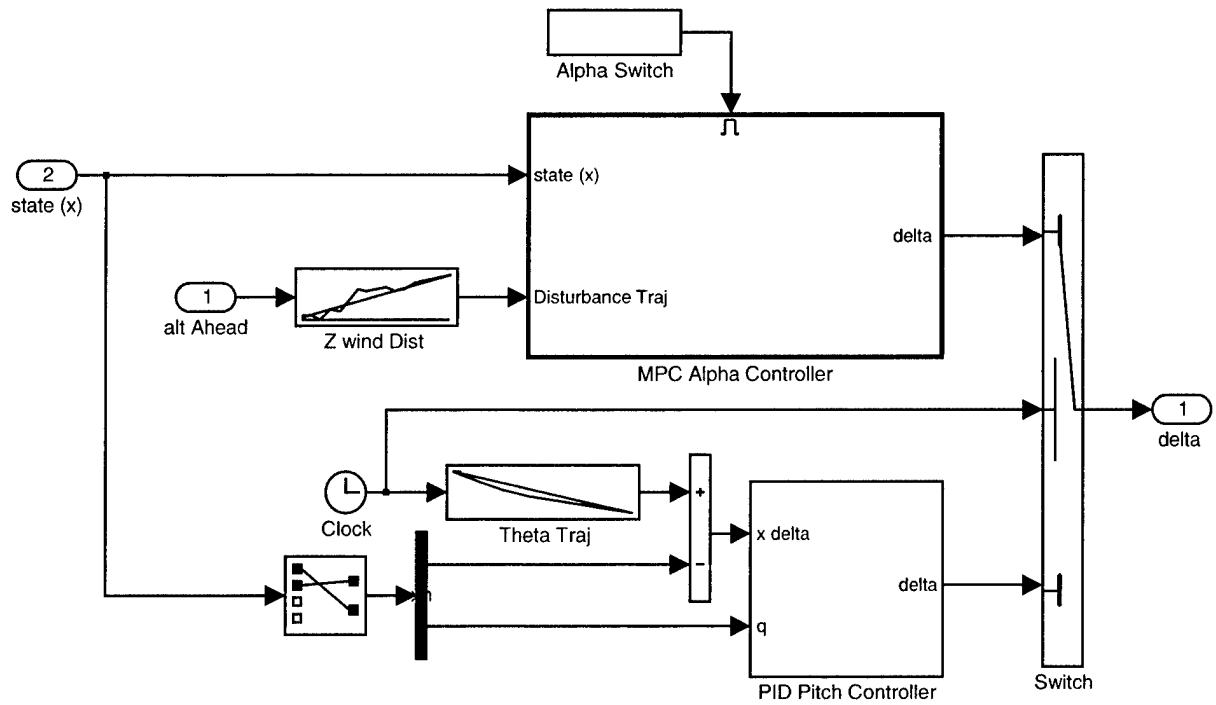
Figure E-3: *NSAS* Model: Pitch Controller

The *NSAS* pitch controller contains both the MPC and PID controllers. The switch transfers control from the PID controller to the MPC controller at a time corresponding to passage of the appropriate Q level. Because the research is only conducted in the pitch plane, which is also the inertial Z plane, the wind disturbance profiles is contained in a look-up table titled *Z wind Dist*. This table produces a vector of disturbances from each altAhead signal, which is then loaded into the MPC controller.
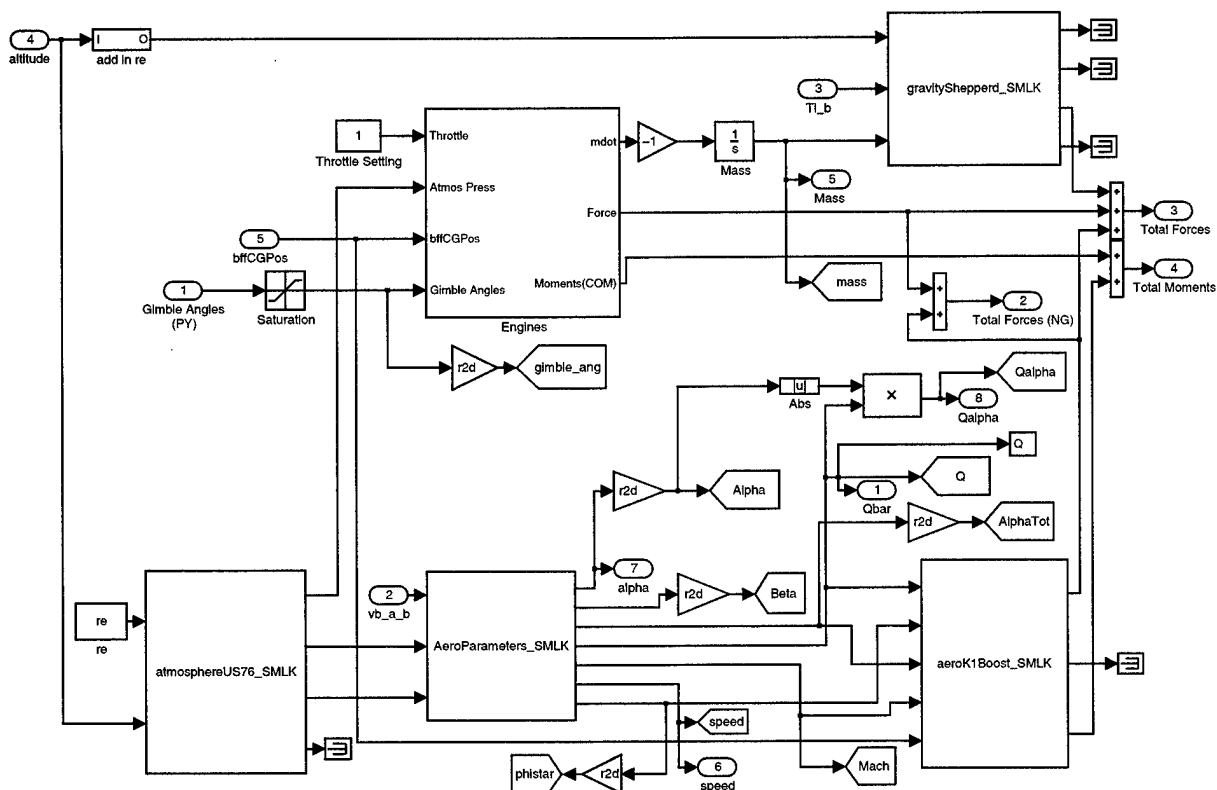
Figure E-4: *NSAS* Model: Forces and Moments

The Forces and Moments subsystem contains all the S-functions required to cal-
culate the forces and moments due to engine forces, aerodynamic forces and gravity.
These S-functions include atmosphereUS76_SMLK, engine_SMLK, aeroK1Boost_SMLK,
gravityShepperd_SMLK and AeroParameters_SMLK. The total forces, neglecting the
effects of gravity are also found, for use in calculating the lateral acceleration of the ve-
hicle. Gains are modelled as triangles and are used most frequently in this subsystem
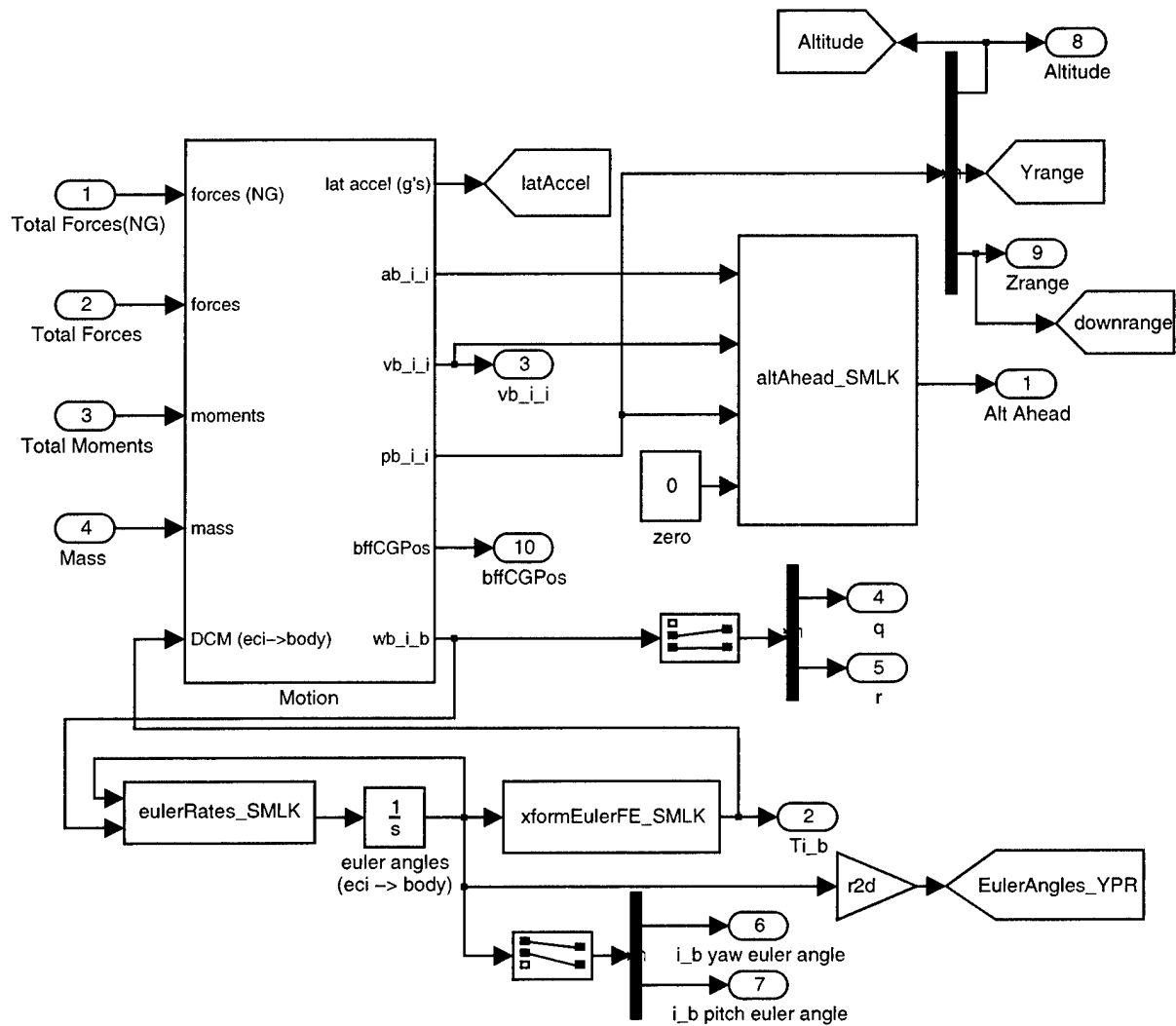to change units from radians to degrees, using the variable $r2d$.

Figure E-5: *NSAS* Model: Vehicle Motion

The Vehicle Motion subsystem contains the S-functions required to propagate the angular position of the vehicle through time, eulerRate_SMLK and a flat earth version of xformEuler_SMLK. It also contains the S-function which produces the altAhead vector of projected altitudes, altAhead_SMLK.
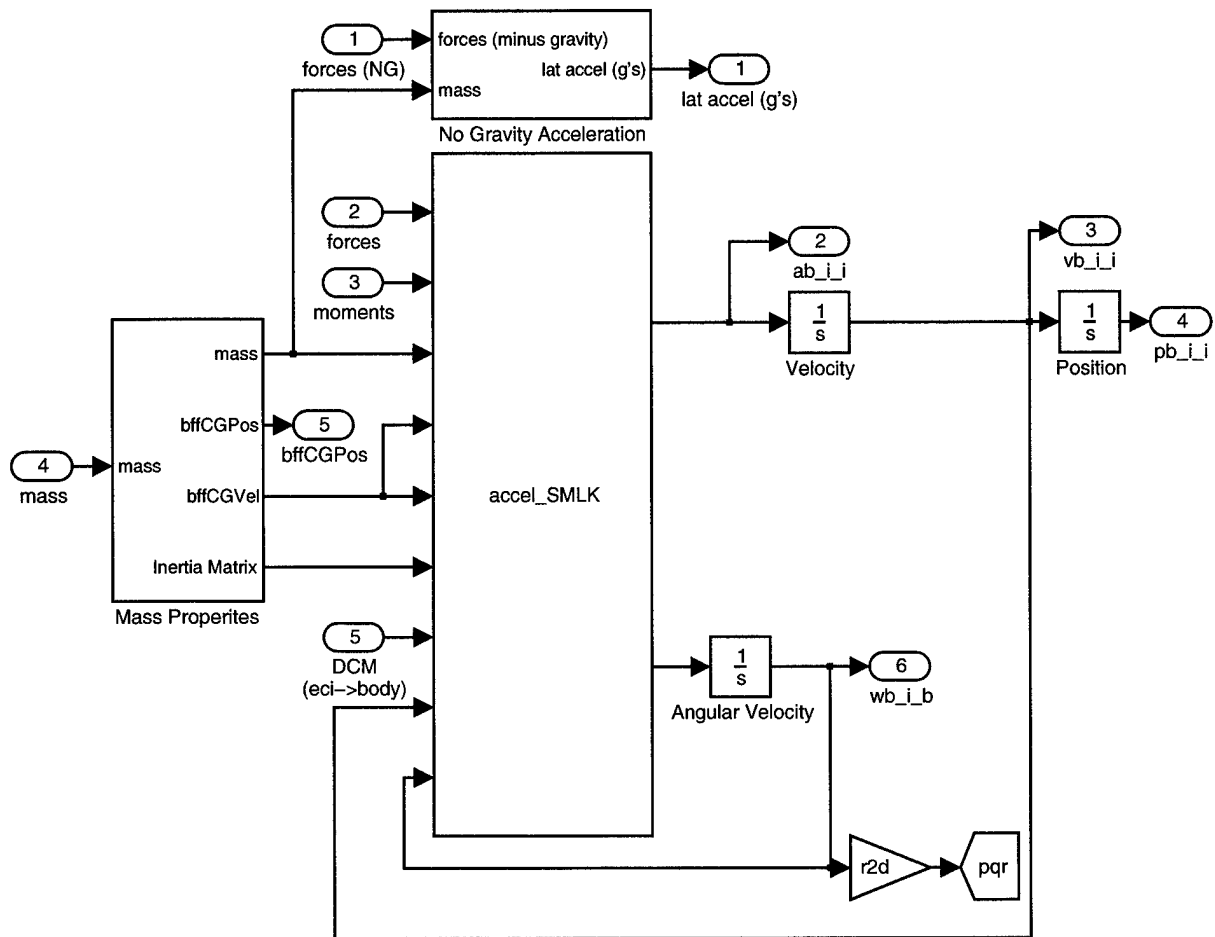
Figure E-6: *NSAS* Model: Motion

The Motion subsystem contains the S-function, accel_SMLK, which calculates linear and rotational accelerations. The integrator blocks used to calculate the linear velocity and position and the rotational velocity are also visible. In addition, the lateral acceleration is calculated using the no-gravity forces calculated in the Forces and Moments subsystem. The Mass Properties subsystem calculates the inertia tensor and CG position as they change over time.
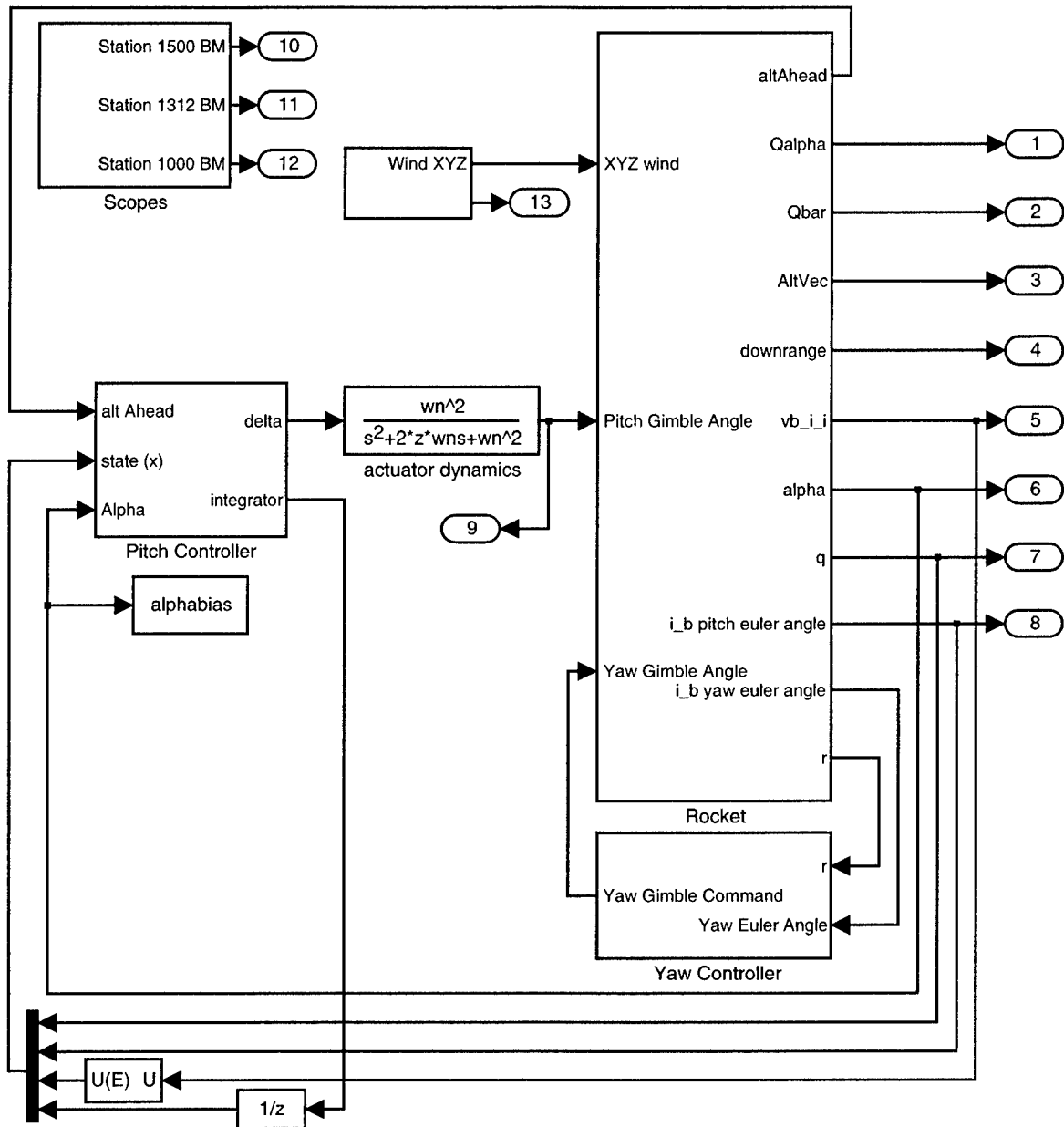
E.3 *ASAS* Model

Figure E-7: *ASAS* Model: Top Level

The *ASAS* simulation top level looks very similar to the *NSAS* simulation top level
with the exception of the additional integrator state which is passed into the pitch
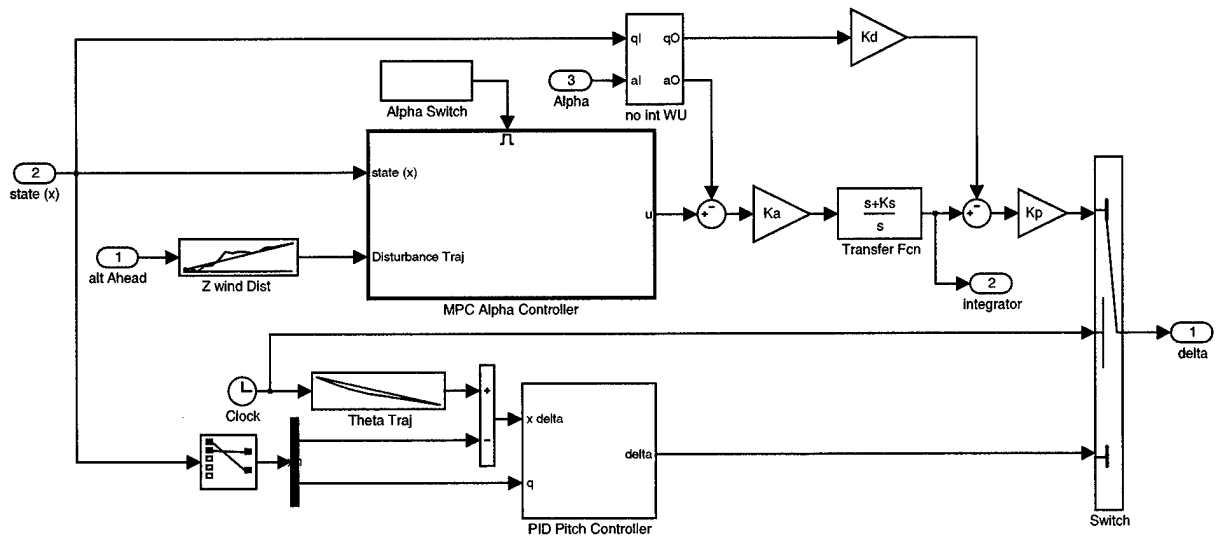controller.

Figure E-8: *ASAS* Model: Pitch Controller

The *ASAS* pitch controller subsystem is very similar to the *NSAS* pitch controller, with the addition of the α SAS between the MPC controller and the output switch. The block labelled *no int WU* is used to eliminate integrator windup during the time when the MPC controller is not operating.
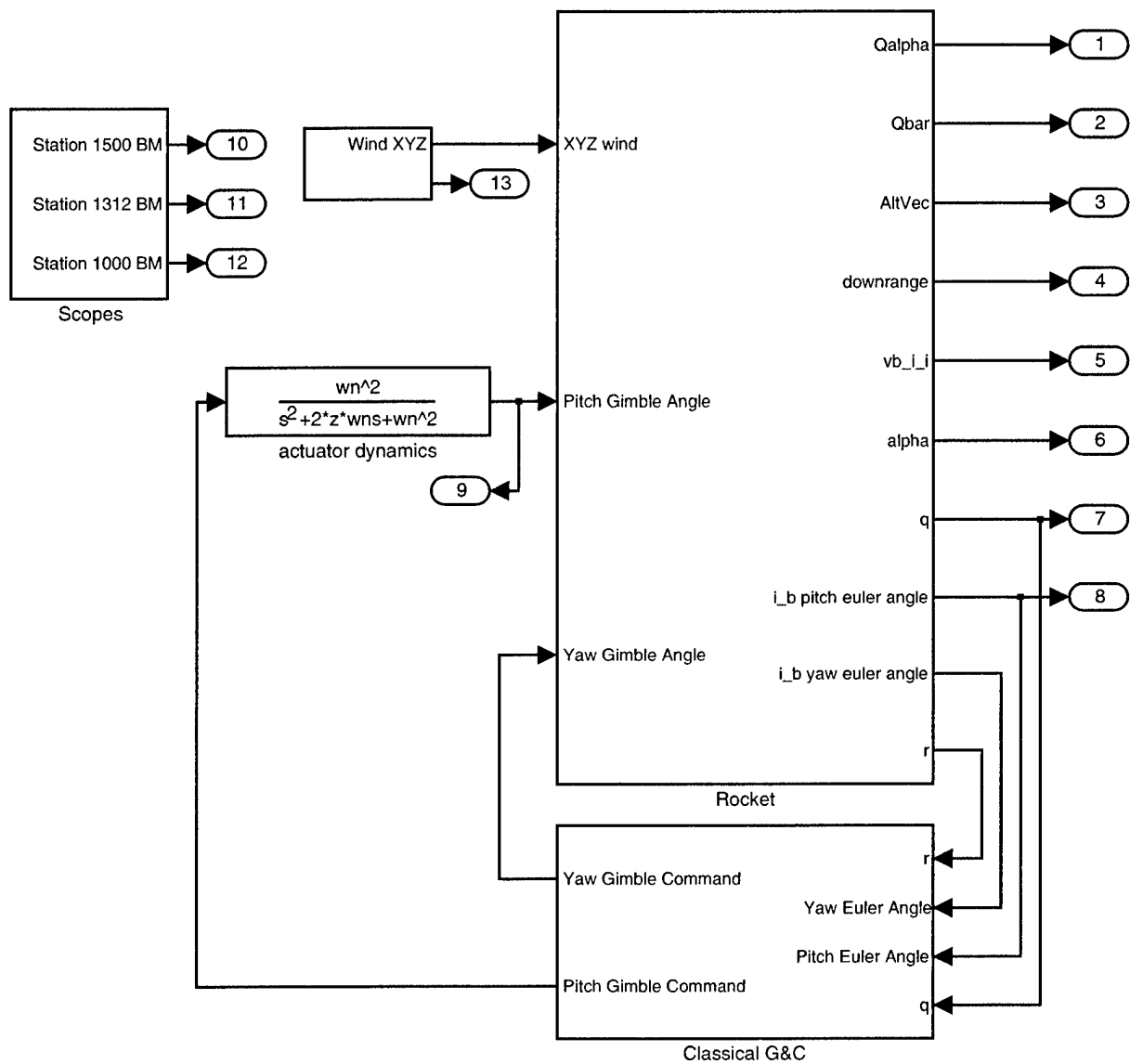
E.4 Baseline Model

Figure E-9: Baseline Model: Top Level

The Baseline simulation shows the common subsystems already discussed in previous sections. The pitch and yaw control are both contained in the subsystem Classical G&C.
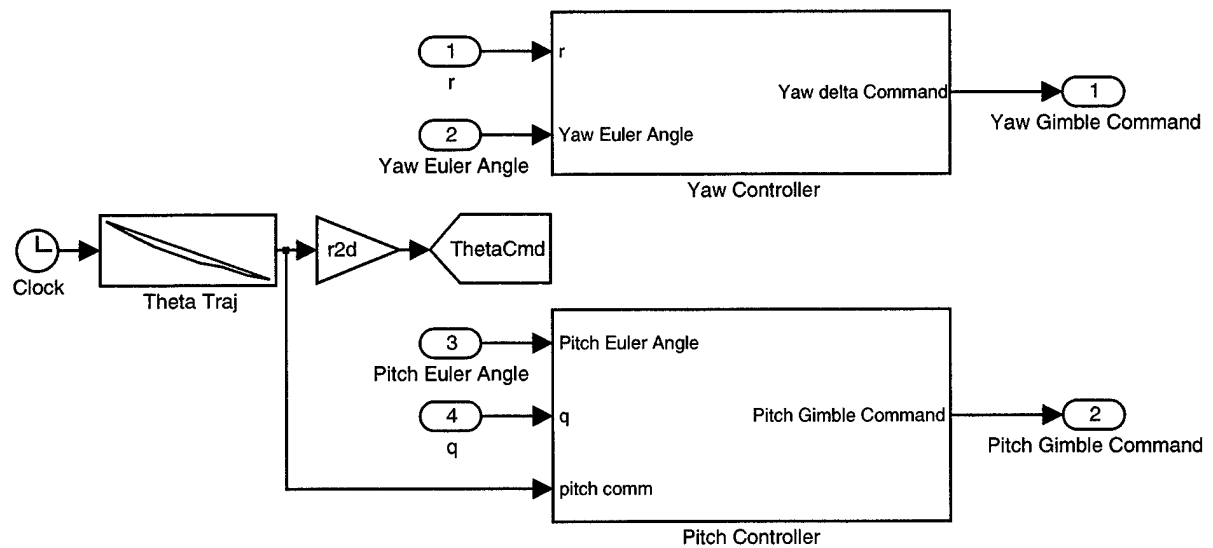
Figure E-10: Baseline Model: Classical G&C

The Baseline pitch controller is shown above. The profile loaded into the look-up table *Theta Traj* is the pitch trajectory calculated by the trajectory designer. The pitch controller follows this commanded trajectory. This controller implements the gain scheduling described in chapter 6.

Bibliography

[1] Bemporad, A., Morari, M., and Ricker, N. L., *The MPC Simulink Library*, Automatic Control Laboratory Physikstrasse 3, 8092 Jurich, Switzerland, June 2001.

[2] Berline, F. and Frank, P. M., *Advances in Model-Based Predictive Control: Design and realization of a MIMO predictive controller for a 3-tank system*, Oxford Science Publications, 1994, pp. 446–457.

[3] Hague, T. N., *An Application of Robust H_2/H_∞ Control Synthesis to Launch Vehicle Ascent*, Master's thesis, Massachusetts Instutute of Technology, June 2000.

[4] Heise, S. A. and Maciejowski, J. M., *Advances in Model-Based Predictive Control: Stability of constrained MBPC using an internal model control structure*, Oxford Science Publications, 1994, pp. 230–244.

[5] Moreno, C. S., *Load Relief Optimization of a First Stage Boost Vehicle Using an Automated Computer Design Tool*, Master's thesis, Massachusetts Instutute of Technology, May 1988.

[6] National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, *U.S. Standard Atmosphere, 1976*, October 1976.

[7] Phillips, C. L. and Nagle, H. T., *Digital Control System Anlysis and Design*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 3rd ed., 1995.

[8] Pines, S., "Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, No. 11, November 1973.

[9] Pines, S. and Austin, G., "Gravitational Acceleration of a Point Mass due to a Rotating Nonspherical Body," Tech. Rep. 69-12, Analytic Mechanics Associates, Inc., May 1969.

[This page intentionally left blank.]